PEARSON EDEXCEL INTERNATIONAL A LEVEL

INFORMATION TECHNOLOGY

Student Book 2

Alan Jarvis
Ian Gibson

Published by Pearson Education Limited, 80 Strand, London, WC2R 0RL
https://www.pearson.com/international-schools

Copies of official specifications for all Edexcel qualifications may be found on the website: https://qualifications.pearson.com

Text © Pearson Education Limited 2024
Development edited by Martin Payne
Copy edited by Just Content Ltd
Proofread by Just Content Ltd
Indexed by
Designed by EMC
Typeset by Straive Ltd
Original illustrations © Pearson Education Limited 2024
Cover design © Pearson Education Limited 2024

Inside front cover: **Shutterstock.com**/Dmitry Lobanov

The right of Ian Gibson and Alan Jarvis to be identified as authors of this work has been asserted by them in accordance with the Copyright, Designs and Patents Act 1988.

First published 2024

27 26 25 24
10 9 8 7 6 5 4 3 2 1

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

ISBN 978 1 292 46807 5

Copyright notice
All rights reserved. No part of this publication may be reproduced in any form or by any means (including photocopying or storing it in any medium by electronic means and whether or not transiently or incidentally to some other use of this publication) without the written permission of the copyright owner, except in accordance with the provisions of the Copyright, Designs and Patents Act 1988 or under the terms of a licence issued by the Copyright Licensing Agency, 5th Floor, Shackleton House, 4 Battlebridge Lane, London, SE1 2HX (www.cla.co.uk). Applications for the copyright owner's written permission should be addressed to the publisher.

Printed in the UK by Bell & Bain Ltd

Endorsement statement
In order to ensure that this resource offers high-quality support for the associated Pearson qualification, it has been through a review process by the awarding body. This process confirms that this resource fully covers the teaching and learning content of the specification or part of a specification at which it is aimed. It also confirms that it demonstrates an appropriate balance between the development of subject skills, knowledge and understanding, in addition to preparation for assessment.

Endorsement does not cover any guidance on assessment activities or processes (e.g. practice questions or advice on how to answer assessment questions), included in the resource nor does it prescribe any particular approach to the teaching or delivery of a related course.

While the publishers have made every attempt to ensure that advice on the qualification and its assessment is accurate, the official specification and associated assessment guidance materials are the only authoritative source of information and should always be referred to for definitive guidance.

Pearson examiners have not contributed to any sections in this resource relevant to examination papers for which they have responsibility.

Examiners will not use endorsed resources as a source of material for any assessment set by Pearson.

Endorsement of a resource does not mean that the resource is required to achieve this Pearson qualification, nor does it mean that it is the only suitable material available to support the qualification, and any resource lists produced by the awarding body shall include this and other appropriate resources.

Acknowledgements
See page XXX

COURSE STRUCTURE IV
ABOUT THIS BOOK V
SPECIFICATION COVERAGE VII
ASSESSMENT OVERVIEW X
INTRODUCTION XII
UNIT 3 1
UNIT 4 128
GLOSSARY 207
INDEX 210

UNIT 3 1

TOPIC 12 MANIPULATING DATA 1

TOPIC 13 ENABLING TECHNOLOGIES 30

TOPIC 14 USING IT SYSTEMS IN ORGANISATIONS 58

TOPIC 15 SYSTEMS DEVELOPMENT 82

TOPIC 16 EMERGING TECHNOLOGIES 109

UNIT 4 128

TOPIC 17 USE OF FEATURES OF DATABASE SOLUTIONS 128

TOPIC 18 RELATIONAL DATABASE CONCEPTS 144

TOPIC 19 DATABASE SOLUTIONS 172

GLOSSARY 207

INDEX 210

ABOUT THIS BOOK

This book is designed for students following the Edexcel International Advanced Level (IAL) IT specification. It covers the second year of the international A level qualification.

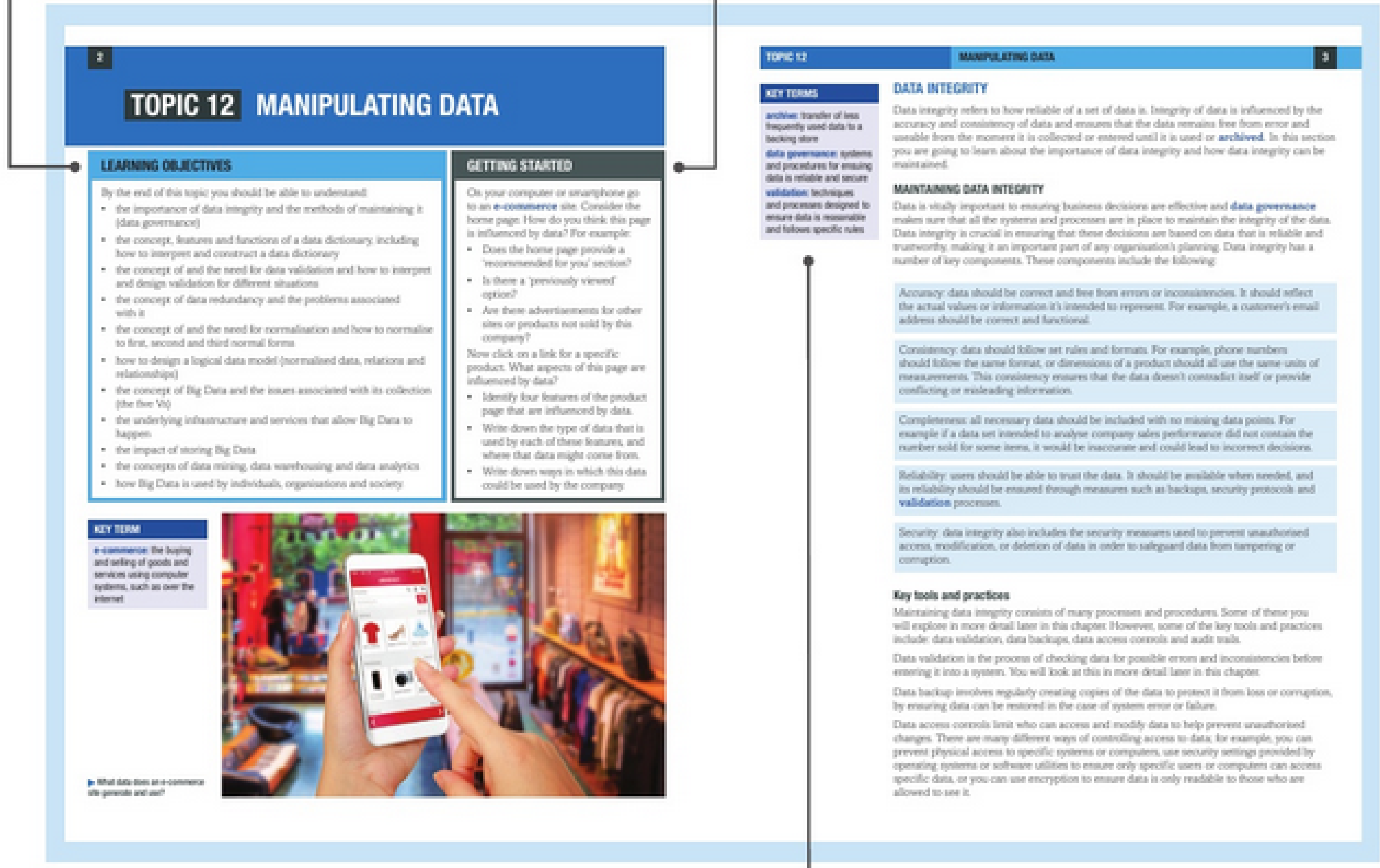
The book has been carefully structured to enable students to develop the knowledge and skills set out in the specification. It is organised into two units. Each of the units is split into several topics. Each topic begins by listing the key learning objectives and includes a getting started activity to introduce the concepts. There is a mix of learning points, key terms and activities throughout. Checkpoint questions and exam practice questions at the end of each chapter help assess understanding of the key learning objectives. Answers for the practice questions can be found online.

The content for Unit 3 is assessed in a two-hour written exam with questions which can include open response questions with short and extended answers. The Unit 4 content is assessed with a three-hour unseen practical and theoretical examination. All responses will be submitted using a word-processed candidate evidence template. Students will be assessed according to four assessment objectives as set out below.

- Demonstrate knowledge and understanding of the concepts of information technology.
- Apply knowledge and understanding of the concepts of information technology.
- Analyse and evaluate IT problems.
- Use analysis and evaluation to design solutions.

Learning objectives
Each topic starts with a list of key learning objectives.

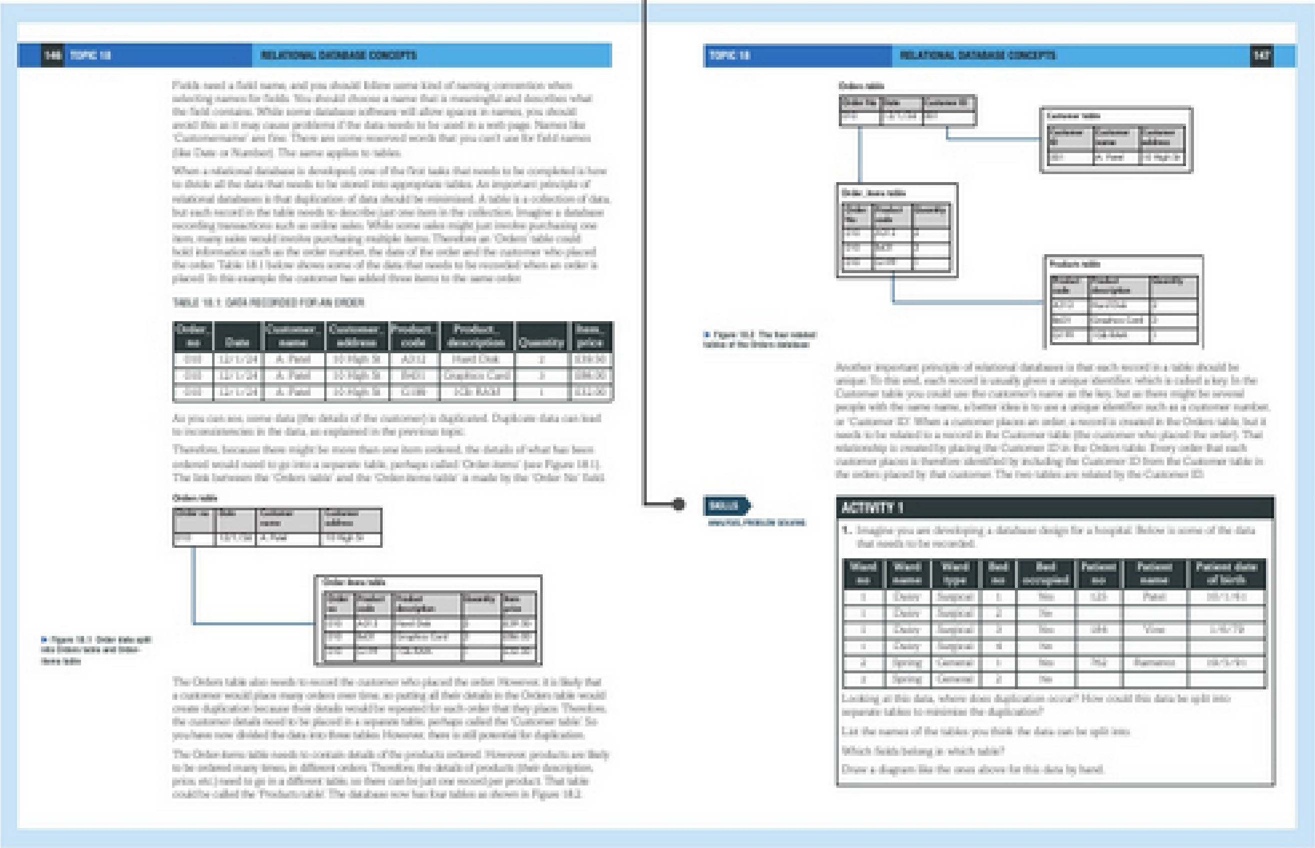
Getting started
Questions are designed to stimulate discussion and use of prior knowledge. These can be tackled as individuals, pairs, groups or the whole class.



Key terms
Key terms are explained as they appear in the text.

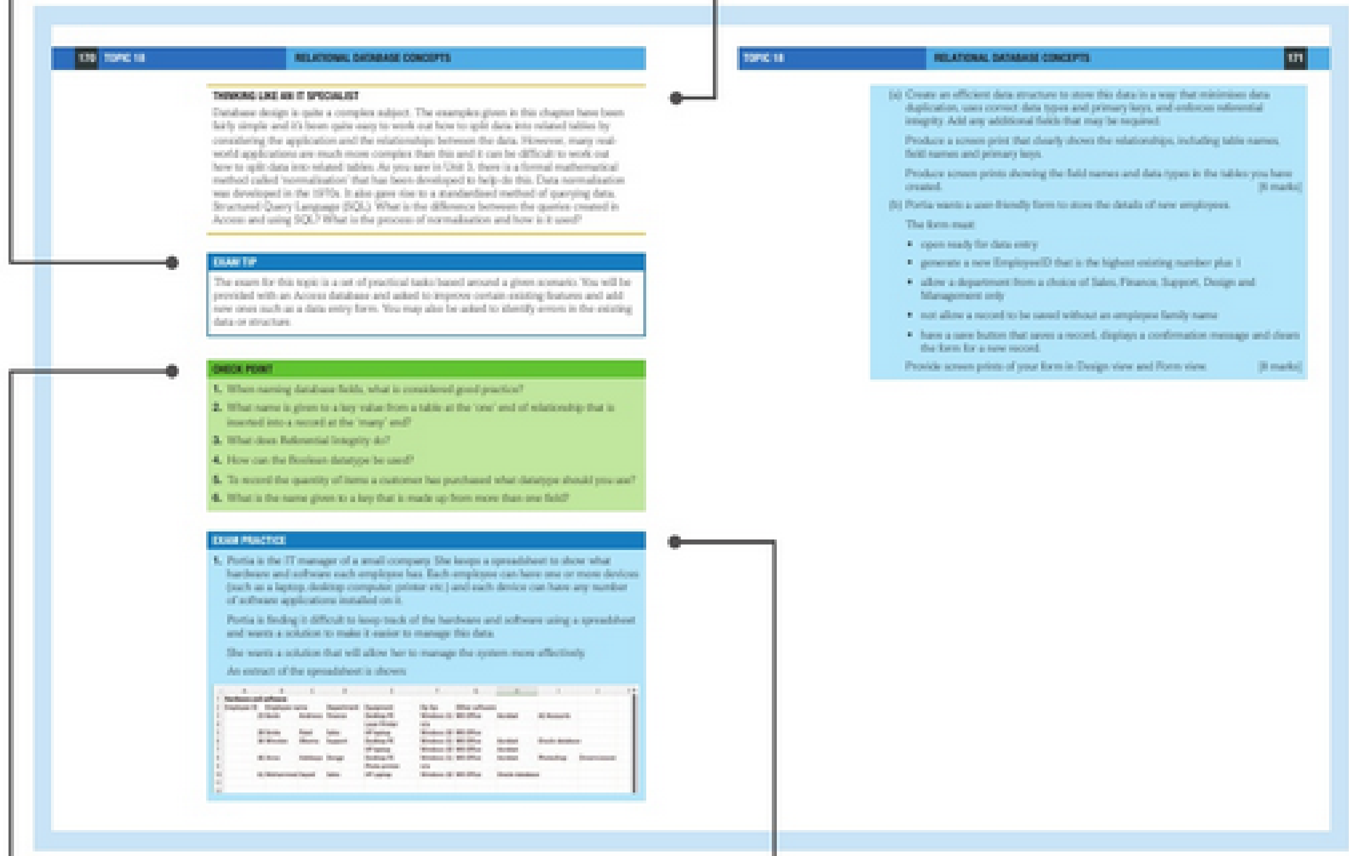
Skills
Activities have been assigned key skills, allowing for a strong focus on particular academic qualities. These transferable skills are highly valued in further study and in the workplace.

Activity
Each topic includes activities to embed understanding.



Exam tip
Tips give practical advice and guidance for exam preparation.

Thinking like an IT specialist
These sections provide the opportunity to explore an aspect of IT in more detail to deepen understanding.



Checkpoint
Questions to check understanding of the key learning points in each chapter. These are NOT exam-style questions.

Exam practice
Exam-style questions are found at the end of each topic. They are tailored to the Pearson Edexcel specification.

SPECIFICATION COVERAGE

UNIT 3

TOPIC 12: MANIPULATING DATA	SPECIFICATION NUMBER	KEY IDEAS
12.1 Data integrity	12.1.1	• The importance of data integrity and the methods of maintaining it (data governance).
	12.1.2; 12.1.3	• The concept, features, and functions of a data dictionary including how to interpret and construct a data dictionary.
	12.1.4; 12.1.5	• The concept of and the need for data validation and how to interpret and design validation for different situations.
12.2 Relational databases	12.2.1	• The concept of data redundancy and the problems associated with it.
	12.2.2; 12.2.3	• The concept of and the need for normalisation and how to normalise to first, second and third normal forms.
	12.2.4	• How to design a logical data model (normalised data, relations and relationships).
12.3 Big Data	12.3.1	• The concept of Big Data and the issues associated with its collection (the five vs).
	12.3.2	• Understand the underlying infrastructure and services which allow Big Data to happen.
	12.3.3	• The impact of storing Big Data.
	12.3.4	• The concepts of data mining, data warehousing and data analytics.
	12.3.5	• How Big Data is used by individuals, organisations and society.

TOPIC 13: ENABLING TECHNOLOGIES	SPECIFICATION NUMBER	KEY IDEAS
13.1 Virtualisation	13.1.1	• What virtualisation is and why it is used.
	13.1.2	• Ways that virtualisation can be achieved.
13.2 Distributed systems	13.2.1	• How distributed systems are used to share data.
	13.2.2	• The issues associated with distributed systems.
13.3 Human computer interaction	13.3.1	• The importance of human computer interaction.
	13.3.2	• Ways that human computer interaction can be implemented.
	13.3.3	• The ergonomic principles associated with human computer interaction.
	13.3.4	• How interfaces are designed to meet the needs and requirements of users.
13.4 Storing and securing data	13.4.1	• How data is stored in the cloud.
	13.4.2	• How data is secured in cloud-based systems.
	13.4.3	• The function of database management systems (DBMS) in controlling access to data.
13.5 Encryption	13.5.1	• The features and functions of symmetric encryption.
	13.5.2	• The features and functions of asymmetric encryption.
	13.5.3	• The features and functions of certificate-based encryption.

TOPIC 14: USING IT SYSTEMS IN ORGANISATIONS	SPECIFICATION NUMBER	KEY IDEAS
14.1 Role of IT systems	14.1.1	• The role of IT systems in organisations.
	14.1.2	• The concept of transaction processing and how it is used in organisations.
	14.1.3	• How and why organisations use customer relationship management (CRM) systems.
	14.1.4	• How management information systems (MIS) are used in organisations.
	14.1.5	• The concept and use of intelligent transportation systems (ITS).
	14.1.6	• The concept of expert systems and how expert systems are used.
14.2 Operational issues	14.2.1; 14.2.2; 14.2.3; 14.2.4; 14.2.5	• Operational issues faced by organisations and the impact on IT systems and users.

TOPIC 15: SYSTEMS DEVELOPMENT	SPECIFICATION NUMBER	KEY IDEAS
15.1 Project management	15.1.1	• The need for project management when developing systems.
	15.1.2	• The characteristics of a successful IT Project.
	15.1.3	• Why project outcomes use SMART targets.
	15.1.4	• How to specify SMART targets.
	15.1.5; 15.1.6	• The tools that are used to manage projects and how to use them.
15.2 Waterfall method	15.2.1	• The concept and features of the waterfall development methodology.
	15.2.2	• The phases of the waterfall method.
15.3 Agile approach	15.3.1	• The concept and features of the agile approach to development methodology.
	15.3.2	• The phases of the agile approach.
	15.3.3	• Activities that take place in an agile approach
15.4 Diagrams used in system design and development	15.4.1	• How to create and interpret information flow diagrams.

TOPIC 16: EMERGING TECHNOLOGIES	SPECIFICATION NUMBER	KEY IDEAS
16.1 Machine learning	16.1.1	• The concepts, features and functions of machine learning.
	16.1.2	• The potential impact and possibilities of machine learning.
16.2 Virtual and augmented reality	16.2.1; 16.2.2	• How and why augmented and virtual reality can be used by individuals and organisations.
16.3 Internet of Things	16.3.1; 16.3.2	• How the Internet of Things (IoT) can affect individuals and organisations.
	16.3.3	• The technology and infrastructure that enables and supports the IoT.
	16.3.4	• Potential security issues relating to the IoT.
	16.3.5	• How to produce high-level plans for IoT solutions.

UNIT 4

TOPIC 17: USE OF FEATURES OF DATABASE SOLUTIONS	SPECIFICATION NUMBER	KEY IDEAS
17.1 Database applications	17.1.1; 17.1.2	• Why databases are used and the typical applications they are used for.
	17.1.3	• The features of database applications.
17.2 User needs	17.2.1; 17.2.2; 17.23	• The different user needs.
17.3 User experience	17.3.1; 17.3.2; 17.33; 17.34	• How the database application can affect the user experience.

TOPIC 18: RELATIONAL DATABASE CONCEPTS	SPECIFICATION NUMBER	KEY IDEAS
18.1 Structuring data	18.1.1	• How to design the structure of a relational database.
	18.1.2	• How to use appropriate data types when structuring data.
	18.1.3	• How to create forms to allow users to input data.
	18.1.4	• How to make input forms user friendly.
18.2 Relational data structures	18.2.1	• Why relational data structures are needed.
	18.2.2	• What are the features and attributes of data relationships.
	18.2.3	• How to evaluate the appropriateness and effectiveness of a relational data structure when given a scenario.
18.3 Data entry and validation techniques	18.3.1	• How to make sure that stored data is suitable for processing.
	18.3.2	• How to use validation techniques to ensure that data is accurate.
	18.3.3	• How construct appropriate and useful error messages.
	18.3.4	• How to improve the quality of data entry.
	18.3.5	• How to evaluate the use of data accuracy techniques.

TOPIC 19: DATABASE SOLUTIONS	SPECIFICATION NUMBER	KEY IDEAS
19.1 Using database software	19.1.1; 19.1.2	• How to create data structures and update data
	19.1.3; 19.1.4; 19.1.5	• How to use database software to create queries and reports.
	19.1.6	• How to import and export data to and from a database.
	19.1.7	• How to create user interfaces for database applications.
	19.1.8	• How to evaluate the effectiveness and appropriateness of database tools
19.2 User interface	19.2.1; 19.2.2; 19.2.3	• How to create data structures and update data
	19.2.4	• How to write macros to automate database functions.
	19.2.5	• How to evaluate the effectiveness and appropriateness of a user interface
19.3 Testing the solution	19.3.1;19.3.2; 19.3.3	• How to test database features.
19.4 Evaluation and enhancement	19.4.1; 19.4.2	• How to evaluate and enhance a database application.

ASSESSMENT OVERVIEW

The following tables give an overview of the assessment for this course. You should study this information closely to help ensure that you are fully prepared for this course and know exactly what to expect in each part of the assessment.

IA2 Unit 3	*Unit code: WIT13/01	
Externally assessed Written examination: 2 hours Availability: June First assessment: June 2020 80 marks	50% of the total IA2	25% of the total IAL
Content overview Students must study **all** of the following topics: • Topic 12: Manipulating data • Topic 13: Enabling technologies • Topic 14: Using IT systems in organisations • Topic 15: Systems development • Topic 16: Emerging technologies. Students will: • demonstrate knowledge and understanding of the concepts of information technology • apply knowledge and understanding of the concepts of information technology • analyse and evaluate IT problems • use analysis and evaluation to design solutions.		
Assessment overview • This paper may include short open-response, open-response and extended open-response questions. • All four assessment objectives are assessed. • Candidates must answer all questions.		

IA2 Unit 4	*Unit code: WIT14/01	
Externally assessed practical examination Written examination: 3 hours Availability: June First assessment: June 2020 80 marks	50% of the total IA2	25% of the total IAL
Content overview Students must study **all** of the following topics: • Topic 17: Use of features of database solutions • Topic 18: Relational database concepts • Topic 19: Database solutions. Students will: • demonstrate knowledge and understanding of the concepts of information technology • apply knowledge and understanding of the concepts of information technology • analyse and evaluate information technology problems • use analysis and evaluation of design solutions.		
Assessment overview • This paper consists of practical and theoretical questions and may include short open-response, open-response and extended open-response questions. • Data files will be provided for some questions. • All responses will be submitted using a word-processed candidate evidence template. • All four assessment objectives are assessed. • Each candidate will need access to a computer during the examination. • Candidates must answer all questions.		

INTRODUCTION

INTRODUCTIONS

Unit 3

Information Technology is integrated with, or supports, almost every aspect of modern life. From communicating and socialising with others, to buying the weekly groceries, to making a decision about where to go on holiday, to completing a class assignment, IT provides systems and data to aid and enhance the tasks that we complete.

By choosing to follow the International A Level course in Information Technology (IT), you are on a journey to equip yourself with the theoretical knowledge and understanding to anaylse the needs of yourself and others and make informed decisions on how IT can be implemented to meet real-world needs. Moreover, you will develop practical skills that will support any future career or study path that you choose, whether it is in the field of IT and computing or in many other areas.

IT is an interesting, engaging and ever-changing area of study which will equip you with analytical, logical and problem-solving skills which will permeate into all areas of your life.

I wish you all the very best with your studies.

Ian

Unit 4

Database technology is an important part of many IT applications, including web-based systems used for a variety of purposes including online shopping, social media and many other systems. Although in many cases the use of a database is not obvious, it is database technology that allows the storage, searching and presenting of data that is essential to the application. In this unit you will look at the different features that database systems have, and the needs of users who will use the systems. You will investigate relational database concepts including how to structure data and create relationships and how to validate data input. Finally, you will learn how to use database software to create user interface features such as queries, menus and reports so that a complete database application can be created.

The assessment for this unit is an exam, which includes practical tasks using database software and theory questions.

Alan

TOPIC 12
MANIPULATING DATA

Data is an important part of the modern world. Nearly every interaction we have with a digital device is driven by data, whether looking at the number of likes and shares that a social media post has received, or the rating reviews of a product we are looking to buy. On a larger scale, data is a valuable resource that helps organisations make informed operational and strategic decisions. This topic will look at how to design and create mechanisms to ensure that data maintains its value as a resource. This topic will also look at the uses of Big Data and the mechanisms involved in collecting, storing and accessing large data sets.

TOPIC 12 MANIPULATING DATA

LEARNING OBJECTIVES

By the end of this topic you should be able to understand:

- the importance of data integrity and the methods of maintaining it (data governance)
- the concept, features and functions of a data dictionary, including how to interpret and construct a data dictionary
- the concept of and the need for data validation and how to interpret and design validation for different situations
- the concept of data redundancy and the problems associated with it
- the concept of and the need for normalisation and how to normalise to first, second and third normal forms
- how to design a logical data model (normalised data, relations and relationships)
- the concept of Big Data and the issues associated with its collection (the five Vs)
- the underlying infrastructure and services that allow Big Data to happen
- the impact of storing Big Data
- the concepts of data mining, data warehousing and data analytics
- how Big Data is used by individuals, organisations and society.

GETTING STARTED

On your computer or smartphone go to an **e-commerce** site. Consider the home page. How do you think this page is influenced by data? For example:

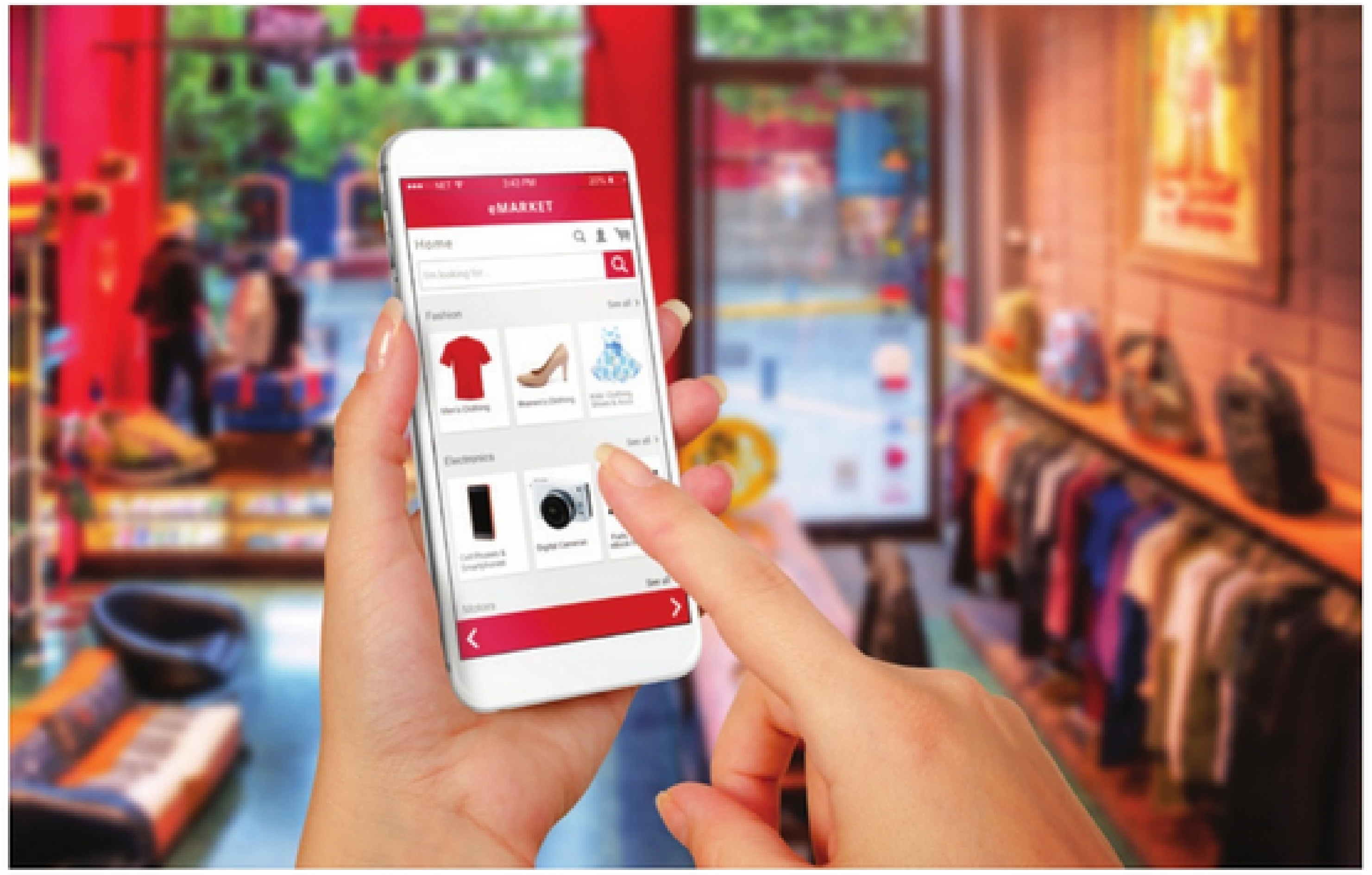
- Does the home page provide a 'recommended for you' section?
- Is there a 'previously viewed' option?
- Are there advertisements for other sites or products not sold by this company?

Now click on a link for a specific product. What aspects of this page are influenced by data?

- Identify four features of the product page that are influenced by data.
- Write down the type of data that is used by each of these features, and where that data might come from.
- Write down ways in which this data could be used by the company.

KEY TERM

e-commerce: the buying and selling of goods and services using computer systems, such as over the internet



▶ What data does an e-commerce site generate and use?

KEY TERMS

archive: transfer of less frequently used data to a backing store

data governance: systems and procedures for ensuing data is reliable and secure

validation: techniques and processes designed to ensure data is reasonable and follows specific rules

DATA INTEGRITY

Data integrity refers to how reliable of a set of data is. Integrity of data is influenced by the accuracy and consistency of data and ensures that the data remains free from error and useable from the moment it is collected or entered until it is used or **archived**. In this section you are going to learn about the importance of data integrity and how data integrity can be maintained.

MAINTAINING DATA INTEGRITY

Data is vitally important to ensuring business decisions are effective and **data governance** makes sure that all the systems and processes are in place to maintain the integrity of the data. Data integrity is crucial in ensuring that these decisions are based on data that is reliable and trustworthy, making it an important part of any organisation's planning. Data integrity has a number of key components. These components include the following:

Accuracy: data should be correct and free from errors or inconsistencies. It should reflect the actual values or information it's intended to represent. For example, a customer's email address should be correct and functional.

Consistency: data should follow set rules and formats. For example, phone numbers should follow the same format, or dimensions of a product should all use the same units of measurements. This consistency ensures that the data doesn't contradict itself or provide conflicting or misleading information.

Completeness: all necessary data should be included with no missing data points. For example if a data set intended to analyse company sales performance did not contain the number sold for some items, it would be inaccurate and could lead to incorrect decisions.

Reliability: users should be able to trust the data. It should be available when needed, and its reliability should be ensured through measures such as backups, security protocols and **validation** processes.

Security: data integrity also includes the security measures used to prevent unauthorised access, modification, or deletion of data in order to safeguard data from tampering or corruption.

Key tools and practices

Maintaining data integrity consists of many processes and procedures. Some of these you will explore in more detail later in this chapter. However, some of the key tools and practices include: data validation, data backups, data access controls and audit trails.

Data validation is the process of checking data for possible errors and inconsistencies before entering it into a system. You will look at this in more detail later in this chapter.

Data backup involves regularly creating copies of the data to protect it from loss or corruption, by ensuring data can be restored in the case of system error or failure.

Data access controls limit who can access and modify data to help prevent unauthorised changes. There are many different ways of controlling access to data; for example, you can prevent physical access to specific systems or computers, use security settings provided by operating systems or software utilities to ensure only specific users or computers can access specific data, or you can use encryption to ensure data is only readable to those who are allowed to see it.

Audit trails involve tracking any changes made to data, and by who, which provides accountability and transparency in case an issue with the data should arise. Audit trails vary depending on the nature of the data, the system being tracked and the specific requirements of the company. However, it is typical for these audits to include:

- a timestamp detailing date and time of the action
- the username (or similar) of the person who performed the action
- the action performed, which would provide details of the specific operation performed on the data (e.g., created, accessed, modified, deleted)
- a record of the specific data element affected by the action
- details of the source of the change, which would include information about the system or application where the action originated.

SKILLS

ANALYSIS, INTELLECTUAL INTEREST AND CURIOSITY

ACTIVITY 1

Data access controls are a key part of ensuring data integrity.

Use the internet to find out more information about the following factors relating to data access control.

- Authentication including:
  - usernames and passwords
  - multi-factor authentication
  - digital certificates.
- Authorisation including:
  - role-based access controls (e.g. admin, management, employee)
  - attribute-based access controls (e.g. user location, time of day, specific device)
  - permission-based control.

You should also explore how different types of access control protect data including:

- preventative measures that block unauthorised access to data upfront (e.g. firewalls, encryption)
- detective methods that monitor and log data access activity for later analysis (e.g. audit trails)
- corrective measures that take action if unauthorised access occurs (e.g. account lockout, alerts).

Add details of what you find during your research to your class notes. Take some time to discuss your findings with a partner. Update your notes to include anything additional you learned from your discussion.

DATA DICTIONARIES

The term 'secure by design' is a fundamental guiding principle in many areas of information technology. Computer programmers use the term to consider how they plan to build and implement their code, but it applies equally to the data that is connected to, and used in, information technology systems. Considering data integrity at the planning stage helps ensure that the data integrity can be established and maintained more easily. One way of doing this is through the creation of a data dictionary.

KEY TERM

metadata: details or information that describe other data to provide deeper meaning or understanding

A data dictionary is like a blueprint or plan for a database or dataset. It provides a centralised and agreed plan, detailing exactly how data will be structured and stored in a computer system. The data dictionary provides **metadata** for the data elements, which helps ensure that all data complies with the required structure, definitions, relationships and attributes required for the specific context.

CONSTRUCTING A DATA DICTIONARY

The data dictionary describes how data will be implemented and helps anyone interacting with the database to understand the structure and properties of the data stored within it. This enables more effective database management and data querying, and ensures data consistency and integrity.

While the exact structure of a data dictionary may vary slightly between organisations, there are some common approaches and content that can be found in most effective data dictionaries.

Typically a data dictionary will include table name, field name, description, data type and length, validation rules and constraints.

- Table name: the name of the specific table within the database that this dictionary applies to. Each table within a database, or group of databases, should have a unique name compared to any other table that it might interact with.
- Field name: the identifier for the specific category of data within a table. For example, 'email address' might be a field within a table that holds customer contact details.
- Description: provides further information about the data contained in it. It should help provide a clear understanding of what the field represents in order to avoid confusion or ambiguity. This is particularly important where a field name is made up from abbreviations or there is potential variation. For example, tel_num would need to ensure that it clarifies it is a telephone number, what type of number (landline or mobile/cell phone). The description may also provide information about the purpose of the field and why it exists in the database.
- Data type: indicates the way in which the data will be stored, which defines the type of operations that the database can perform on the data. For example, storing the data as an integer would allow mathematical operations to be performed on the data, whereas storing as a string data type allows a range of text-based characters to be stored. Table 12.1 provides details of common data types used in databases.
- Length/size: defines the maximum length or size of the data to be input. Limiting the size helps reduce the chance of rogue data being added.
- Validation rules: provides restrictions to the data that can be entered to ensure any data that is entered is reasonable. While validation does not prevent data from being incorrect, it does reduce the potential scope of any errors in the data.
- Constraints: defines any additional rules or restrictions applied to the data (such as being a primary key, not null, or how it relates to other data and/or tables). You will explore some of these concepts in more detail in the relational database section of this chapter.

KEY TERM

structured query language (SQL): a widely used programming language used for manipulating and maintaining data

Table 12.1 shows some common data types used in **structured query language (SQL)** databases. Not all databases use these exact data types, although many are common, other data systems will have similar compatible data types.

TABLE 12.1: SOME COMMON DATA TYPES USED IN SQL DATABASES

Common SQL datatypes		
	Data type	**Description**
Integer data types	INT	Stores whole numbers within a specific range (usually -2147483648 to 2147483647).
	SMALLINT	Used for smaller integers when a smaller range would be needed. Defining a smaller range saves on storage space. Range for SMALLINT is: -32768 to 32767 or 0 to 65535
	BIGINT	For very large positive and negative numbers. Used when the numbers to be stored would exceed the range of the INT datatype.
	TINYINT	Used for small numbers. Typically 0 to 255
Decimal types	DECIMAL	Stores precise decimal numbers with a specified precision and scale (e.g., DECIMAL(10,2) for up to 10 digits, 2 of which are decimal places).
	FLOAT and REAL	For floating-point numbers (approximate values with varying precision).
Date and time types	DATE	Stores date values in the format YYYY-MM-DD (e.g. 2024-01-10).
	TIME	Stores time values in the format HH:MM:SS (e.g. 13:00:00).
	DATETIME	Combines date and time (e.g. 2024-01-10 13:00:00).
	TIMESTAMP	Similar to DATETIME, often used for tracking record creation or modification timestamps.
Character string types	CHAR	Text-based string with a fixed length e.g. CHAR(8) for an 8-character string.
	VARCHAR	Used for variable length character strings. Is space efficient when working with strings that will vary in length, such as names or addresses.
	TEXT	For larger blocks of text.
	NCHAR and NVARCHAR	Unicode compatible – used for storing multilingual text.
Other types	BINARY and VARBINARY	Stores binary data such as images, audio/video or other files.
	BOOLEAN	Stores TRUE or FALSE values only.
	JSON	Not supported by all databases or systems, but is a data format that is increasingly used to pass data between systems. Used if a database will be passing data to and receiving data from other systems.

A simple data dictionary that a sports club might use when building a database to store its membership details could look like Table 12.2.

TABLE 12.2: A SIMPLE DATA DICTIONARY FOR MEMBERSHIP DETAILS

Field name	Description	Data type	Length
Member_name	The member's full name	Varchar	20
Member_address	The member's full address	Varchar	50
Phone_no	The member's contact phone number	Int	15
Email_addr	The member's email address	Varchar	20

The data type Varchar is used for the name and address field because the club doesn't know in advance how long these fields will be. Even though the length is given for each field, this is the maximum length as the database will only assign the space required for each name and address. This makes the database smaller, and therefore more efficient.

Defining the fields in the data dictionary before starting to build the database, forms and reports makes sure that everything is consistent throughout.

SKILLS

PROBLEM SOLVING, ANALYSIS

ACTIVITY 2

Gabrielle owns a company that offers dog-walking services. Her company is growing rapidly and she has realised that she needs a database to help support her administrative tasks.

She needs to hold this data about each of her clients:

- dog's name
- owner's given name
- owner's family name
- address
- telephone number
- days to be walked
- time of walking appointment
- length of walk (hours and minutes)
- specific medical/dietary needs of dog
- registered with veterinarian (Yes/No)
- name of vet
- contact number of vet

Create a basic data dictionary that will inform the creation of a data table for Gabrielle's business.

Use these column headings to create your data dictionary:

Field name	Description	Data type	Length/size

DATA VALIDATION

Reducing the chance of errors in stored data helps to improve the integrity of the data. Data validation implements rules and procedures to attempt to reduce the chance of errors occurring. It is important to realise that data validation does not ensure that data is completely error free. However, it does ensure that data entered into a particular field follows pre-set rules that reduce the number and types of error that can occur.

Data validation ensures that data is 'reasonable', that is to say that it limits or categorises the type of data that can be entered into a field so that data is more likely to be reliable and complete.

Data validation typically follows three phases (see Figure 12.1 on next page).

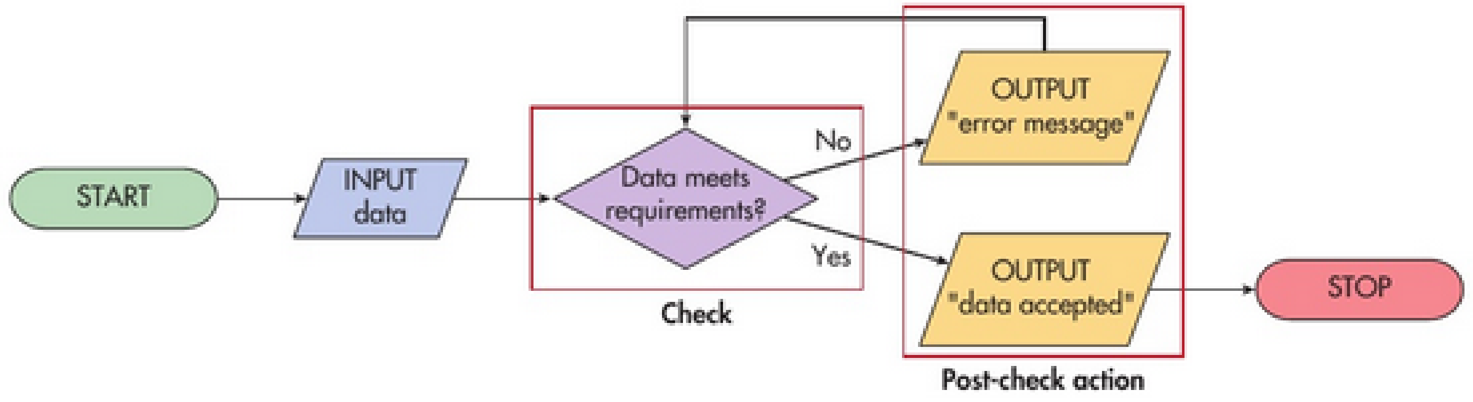
KEY TERM

data migration: extracting data from an existing dataset and transferring it to another

(User) input: first the data must be input into the system. This is often where the user types in some data, however this can also apply to automated data entry and **data migration**.

Check: the system compares the data against the predefined rules, which will contain one or more conditions that the data must satisfy. Depending on how the check is resolved, the system logic will dictate which 'post-check action' to take.

Post-check action: once the check has taken place the computer must perform an action in response to the data entered. If the data entered does not satisfy the rules, typically the system will output an error message explaining the issue and prompt the user to re-enter the data. If the data satisfies the rules, then the system will proceed to the next data item or action. It is also good practice to provide the user with confirmation that data has been accepted. For example, on a form on a website, a green tick might appear next to the data entry field.



▶ Figure 12.1 The main stages of a data validation check

DATA VALIDATION RULES

Effective validation rules will limit the scope of the data to be entered, so it is vital to select the correct type of data validation for each specific use case. Data validation rules can be categorised into seven main types. Most of these can then be designed to apply specific rules and parameters to the data.

1 Presence check

This is checking that data has been entered into a data field. It ensures mandatory fields are not left blank. For example, the customer's name and delivery address when making a purchase online.

2 Range check

This ensures that data entered falls between two accepted values. For example, checking that an input for the length of time required to leave a car in a car park is between 1 and 6 hours.

3 Lookup

This checks the data entered against a pre-existing reference list or table. The data used for the check is typically stored outside of the data table. For example, it may be used to check that a product code typed into an order form matches a product code in the stock table. Another use could be to check that when a user is signing up for an account, the email they provide isn't already being used by another account.

4 List

Similar to a lookup check in that it compares the entered data to a list of predefined acceptable entries; however, the list differs slightly in that it is typically used for smaller lists of data. For example, it could be used on a website when purchasing a train ticket and restricts the options for the type of ticket to "Child", "Adult" or "Family".

5 Length

This checks the number of characters entered to ensure that the data entry meets the minimum and maximum length requirements. For example, a password may have to be more than 8 characters long, or a username might be restricted to a maximum length of 20 characters.

6 Format check

This checks that the data entered follows a specific format and/or contains required characters. For example, ensuring that a date of birth is entered using the format DD/MM/YYYY, or an email address contains the '@' symbol and a valid domain name (e.g. .com, .org).

7 Check digit

A check digit is added to the end of a string of numbers to detect errors such as incorrect data entry or transcription errors. The number that is added is calculated using a predefined algorithm. When the number is entered into a system, the system recalculates the check digit using the same algorithm. If the calculated check digit matches the one provided, it suggests that the entered number is likely to be accurate. Otherwise, it signals a potential error in data entry or transmission. Typical uses include the ISBN for a book and credit card numbers.

Table 12.3 shows the logic for the data validation rules for each of the listed types. In each case, if a check returns True it means that the validation rule has been passed, and the data is considered valid. A return of False means the test has failed and the data is not valid.

TABLE 12.3: HOW MIGHT THESE RULES BE APPLIED IN REAL-WORLD SITUATIONS?

Validation rule type **Rule logic**

Presence

```
IF LENGTH (user_input) == 0:
    RETURN False
ELSE
    RETURN True
```

Range

```
IF value < minimum OR value > maximum:
    RETURN False
ELSE:
    RETURN True
```

Lookup

```
IMPORT data FROM TABLE_X

GET user_input FROM KEYBOARD

IF user_input IN data:
    RETURN True
ELSE:
    RETURN False
```

List

```
Options = [option 1, option 2, option 3]

GET user_input FROM KEYBOARD

IF user_input IN options:
    RETURN True
ELSE:
    RETURN False
```

Length

```
IF LENGTH (user_input) > minimum AND LENGTH (user_input) < maximum:
    RETURN True
ELSE
    RETURN False
```

Format

```
IF required_character IS IN user_input:
    RETURN True
ELSE
    RETURN False
```

SKILLS

CRITICAL THINKING, PROBLEM SOLVING

ACTIVITY 3

Pika owns a company that organises water sports events, such as surfing and kayaking.

He requires a data table to store the information about people who enter the events.

Copy and complete this table to create a data dictionary for the data table.

Table	Field name	Description	Datatype	Length	Validation rule	Constraints
Event_24	Entry_num	Competitors entry number. Used to identify booking and competitor in race	CHAR(10)	10	Presence check Length >0 and length <11	Must be unique. Primary key
Event_24	Entry_date	Date entry booking was made				
Event_24	Date of birth					
Event_24	Name					
Event_24	Address					
Event_24	Event entered					

What other fields might be included?

Pika has data from other events stored in separate tables. How might he use that data to help with this data table?

RELATIONAL DATABASES

A relational database, like any form of database, stores data in an organised and structured manner. One of the key features of a relational database is how it links data (creates relationships) between items of data in separate tables. Relational databases are useful when you need to organise a large amount of different but related data.

An e-commerce site, for example, needs to store information about customers, the products it sells, suppliers and keep records of sales. Storing all this information in a single data table would become complex and can lead to errors and **data redundancy**. A relational database would allow you to store this data in separate tables and use and combine data from different tables when needed. For example, when a new customer registers, their information would be added to the table that holds customer information. When they make a purchase, however, data would be needed from both the customer table and the product table; the database would also then need to store details of this order.

KEY TERM

data redundancy: the unnecessary replication of data in a database

SKILLS

REASONING/ARGUMENTATION

ACTIVITY 4

In the introduction to this section there was the example of an e-commerce site, which highlighted a situation where a lot of separate but related data would be required.

Working with a partner:

- discuss different situations that would require lots of data and may require a relational database
- explore what data would be required in the situations that you identify, and why that data would be needed.

DATA REDUNDANCY

A key issue that you should try to avoid when storing data is that of data redundancy. Data redundancy occurs when data is replicated unnecessarily in multiple locations. For example, a company may have customer information stored in multiple databases; it might have separate databases for different departments, such as sales, marketing and customer service. If each database stores the same customer information (name, address, phone number), that is data redundancy.

It should be noted at this point that sometimes the term 'data redundancy' can also be used to refer to intentional duplication of data, such as a backup. While this form of data redundancy can offer some benefits, such as improved availability and disaster recovery, in this section the term is used to refer to the unintentional replication of data.

Data redundancy can cause a number of problems, including increased storage costs, data inconsistency and maintenance implications.

Increased storage costs

Storing the same data multiple times takes up more storage space. As the data set grows, the redundant data makes the data set grow exponentially resulting in much higher storage costs, especially for large datasets.

Data inconsistency

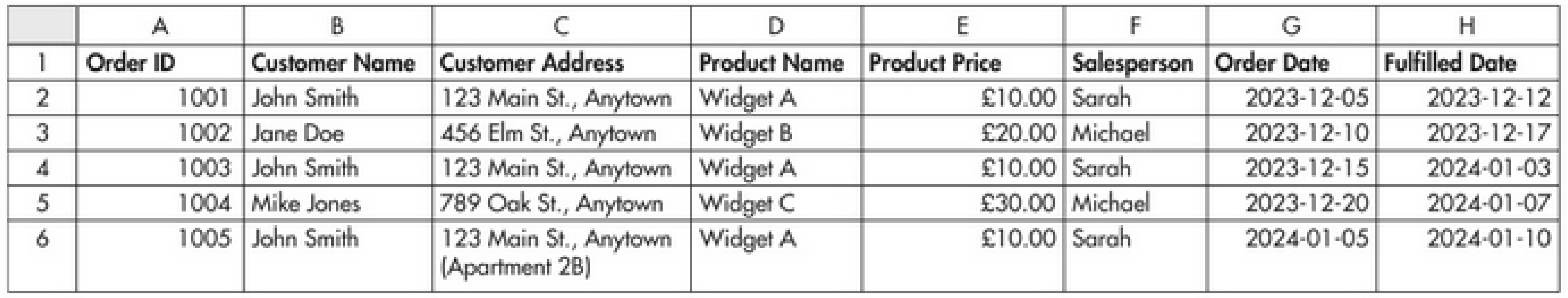
If the same data is updated in different locations but the updates aren't synchronised, inconsistencies can arise, leading to confusion or errors. For example, if a company keeps the customer information in multiple different locations, and that customer moves house, if their address is only updated in one of the data tables then this could cause issues for the company and the customer.

Maintenance implications

Managing and updating redundant data takes time and effort, which could be better spent on other tasks.

Figure 12.2 shows an example of a small data set that contains redundant data. What issues might occur due to the data redundancy in this data set?

▼ Figure 12.2 A data set with redundancy



	A	B	C	D	E	F	G	H
1	Order ID	Customer Name	Customer Address	Product Name	Product Price	Salesperson	Order Date	Fulfilled Date
2	1001	John Smith	123 Main St., Anytown	Widget A	£10.00	Sarah	2023-12-05	2023-12-12
3	1002	Jane Doe	456 Elm St., Anytown	Widget B	£20.00	Michael	2023-12-10	2023-12-17
4	1003	John Smith	123 Main St., Anytown	Widget A	£10.00	Sarah	2023-12-15	2024-01-03
5	1004	Mike Jones	789 Oak St., Anytown	Widget C	£30.00	Michael	2023-12-20	2024-01-07
6	1005	John Smith	123 Main St., Anytown (Apartment 2B)	Widget A	£10.00	Sarah	2024-01-05	2024-01-10

DATA NORMALISATION

To reduce redundancy, data goes through a process called normalisation. During normalisation, complex data tables are broken down in to successively smaller and simpler tables. The aims are to ensure that repeated data is removed, and to create relationships between the tables where a process is reliant on data from multiple areas, such as in the e-commerce example mentioned in the data redundancy section.

There are three main stages to normalisation, referred to as first, second and third normal form. At each stage, stricter rules are applied to the data and redundancy is reduced.

First normal form (1NF)

The aim of first normal form is to remove any repeating groups of data in individual tables, and ensure that each column represents a single attribute. When creating a table in first normal form, your table should meet these requirements:

Each cell should contain single, indivisible values.

All entries in a column must be of the same data type.

Each column should have a unique name.

A **primary key** should be used to identify each **record**.

Second normal form (2NF)

In second normal form, any sets of data that apply to multiple records should be placed into separate tables. For example, in a database that stores information about films and TV programmes, an actor may appear in many different TV programmes and films, so it would be sensible to have a table that contains the names and personal details of the actors (e.g. name, date of birth, etc.). You could then create a separate table for the information that is unique to the films and TV programmes (e.g. release date, run time, genre). Each entry in each table would have its own primary key, which could then be linked using a **foreign key**. See the worked example/case study on page xx for a further example.

When creating a table in second normal form your table should meet these requirements:

First normal form must have been satisfied.

There are no partial dependencies.

Third normal form (3NF)

In third normal form any data in a row that doesn't depend on a key should be eliminated and placed in its own table. While this may not always be possible, or even necessary, it is something that should be aimed for.

For example, in a table that contains customer addresses, there may be many addresses based in the same city. There may be times when the city has an impact on other data; this is called a transitive dependency. In this case the city data item may be used to dictate which distribution centre the ordered item should be dispatched from, or which delivery service to use. In order to remove the transitive dependency "city" would be given a unique ID and placed in a separate table and further relationships would be generated. If, however, the city name is only ever used in association with the customer, such as just printing a shipping label or mailing a letter then no further normalisation would be required.

KEY TERMS

foreign key: a column or columns in a table that links to unique data in another table. This is often a primary key in the other table

primary key: a code, number or name that uniquely identifies a single record in a database

record: an individual row in a data table that contains data about a single person, item or object

To be in third normal form the data must meet these requirements:

The requirements for second normal form must have been satisfied.

Transitive dependencies have been removed.

CASE STUDY: DARCEY'S CAKES – NORMALISATION WORKED EXAMPLE

Darcey runs a small but growing specialist bakery. The company specialises in making celebration cakes for weddings, birthdays and other events.

Stage 1 – Un-normalised data

Currently Darcey keeps records of cake orders on a spreadsheet (Figure 12.3). The data is not normalised.

Customer_name	Address	Cake_type	Size	Cake_description	Deposit_paid	Total cost	Baker
Italia Acqua	123 Main Street, London, SW1A 1AA	Birthday	Small	2 tier, yellow frosting, "Happy 18th Birthday"	Yes	£95	John Simpson
Slavko Newport	789 Pine Close, Birmingham, B13 8JH	Wedding	Large	3 tier, 1x fruit, 2x sponge, Wedding Design 3c	£87.50	£350	Mariami Kaur
Khasan Cai	456 Oak Avenue, Manchester, M2 4WD	Birthday	Large	Chocolate, Novelty Design – Car	No	£150	John Simpson
Khasan Cai	456 Oak Avenue, Manchester, M2 4WD	Birthday	Medium	2 tier, vanilla sponge	Yes	£110	Gerwas McGregor

▲ Figure 12.3 Data does not conform to first normal form (1NF)

As you can see the data does not conform to 1NF:

✗ Each cell should contain single, indivisible values.

✗ All entries in a column must be of the same data type.

✓ Each column should have a unique name.

✗ A primary key should be used to identify each record.

Stage 2 – first normal form (1NF)

The data must first be restructured to ensure that it meets the rules of 1NF (Figure 12.4).

Customer_ID	Customer_name	Address line 1	Town_city	Postcode	Cake_type	Size	Cake_description	Deposit paid?	Deposit_value	Total_cost	Remaining	Baker
Cust_00001	Italia Acqua	123 Main Street	London	SW1A 1AA	Birthday	Small	2 tier, yellow frosting, "Happy 18th Birthday"	Yes	£23.75	£95.00	£71.25	John Simpson
Cust_00002	Slavko Newport	789 Pine Close	Birmingham	B13 8JH	Wedding	Large	3 tier, 1x fruit, 2x sponge, Wedding Design 3c	Yes	£87.50	£350.00	£262.50	Mariami Kaur
Cust_00003	Khasan Cai	456 Oak Avenue	Manchester	M2 4WD	Birthday	Large	Chocolate, Novelty Design – Car	No	£37.50	£150.00	£112.50	John Simpson
Cust_00003	Khasan Cai	456 Oak Avenue	Manchester	M2 4WD	Birthday	Medium	2 tier, vanilla sponge	Yes	£27.50	£110.00	£82.50	Gerwas McGregor

▲ Figure 12.4 Data table in first normal form (1NF)

As you can see the data now conforms to 1NF:

✓ Each cell should contain single, indivisible values.

The address has been broken down in to separate fields so that the address line, town and postcode can be used separately. The data is said to be **atomic**.

The Customer_name field has not been broken down in this example, however if the data was to be used in a way where first name and surname would be used separately, then this would need to be separated to become atomic.

KEY TERM

atomic data: data that represents a single attribute and cannot be broken down further

✓ All entries in a column must be of the same data type.

The Deposit_paid field has been made consistent and it now only uses the Boolean data type.

The additional fields have been added to make use of this data. The Deposit_value, and Remaining could be automatically calculated based on the Total_cost value

✓ Each column should have a unique name.

✓ A primary key should be used to identify each record.

Each customer is now associated with a Customer_ID. Although this satisfies 1NF, the table still includes some redundancy. Because the same customer has made more than one purchase, they appear in the list multiple times, as does their ID number. In the next stage we will look at how this can be rectified.

Stage 3 – second normal form (2NF)

The data is then split into separate tables (customers, cakes, orders).

Customer table: see Figure 12.5.

Customer_ID	Customer_name	Address_line 1	Town_city	Postcode
Cust_00001	Italia Acqua	123 Main Street	London	SW1A 1AA
Cust_00002	Slavko Newport	789 Pine Close	Birmingham	B13 8JH
Cust_00003	Khasan Cai	456 Oak Avenue	Manchester	M2 4WD

▲ Figure 12.5 Customer data table in second normal form (2NF)

Each customer appears only once in the customer table – they each have a Customer_ID, which is the primary key for this table. The primary key will have the constraint UNIQUE, so that it cannot be duplicated.

Cakes table: see Figure 12.6.

Cake_ID	Cake_type	Size
Birth_sml_2t	Birthday	Small
wed_lrg_3t	Wedding	Large
birth_lrg_2t	Birthday	Large
birth_med_2t	Birthday	Medium

▲ Figure 12.6 Cakes data table in second normal form (2NF)

A cakes table has been created. Cake_ID is the primary key for this field. In this example there would be a separate record for each cake and size combination, this would allow Darcey to expand the list of cakes without restructuring the data.

Order table: see Figure 12.7.

Order_ID	Customer_ID	Cake_ID	Cake_description	Deposit_paid	Deposit_value	Total_cost	Remaining	Baker
DC_000001	Cust_00001	Birth_sml_2t	2 tier, yellow frosting, "Happy 18th Birthday"	Yes	£23.75	£95.00	£71.25	John Simpson
DC_000002	Cust_00002	wed_lrg_3t	3 tier, 1x fruit, 2x sponge, Wedding Design 3c	Yes	£87.50	£350.00	£262.50	Mariami Kaur
DC_000003	Cust_00003	birth_lrg_2t	Chocolate, Novelty Design – Car	No	£37.50	£150.00	£112.50	John Simpson
DC_000004	Cust_00003	birth_med_2t	2 tier, vanilla sponge	Yes	£27.50	£110.00	£82.50	Gerwas McGregor

▲ Figure 12.7 Order data table in second normal form (2NF)

Each record in the order table has a unique Order_ID, which is the primary key for this field.

Customer_ID and Cake ID, which are primary keys in their own tables, are also foreign keys in this table. This allows the data to be linked between the tables.

The data now satisfies the requirements for 2NF:

✓ First normal form must have been satisfied.

✓ There are no partial dependencies.

Stage 4 – third normal form (3NF)

The data in the cake database would be useable and present few issues in its current state (2NF). However, the data could be further normalised into 3NF. For example, currently the baker name denotes which baker will make the cake. If the bakery only uses this information to assign a daily job list, they may be happy adding the name of the baker manually each time an order list is created. However, if they wanted to have other data stored about each baker, such as salary, contact information, quality assurance checks, it would be sensible to create an additional table to hold the data related to the bakers.

Also, if the price of each cake is standard then it may be more appropriate to store that data in the cake table.

The customer table remains unaltered as it was already fully normalised.

Here is the data in third normal form (3NF) in the Customer table (Figure 12.8).

Customer_ID	Customer_name	Address_line 1	Town_city	Postcode
Cust_00001	Italia Acqua	123 Main Street	London	SW1A 1AA
Cust_00002	Slavko Newport	789 Pine Close	Birmingham	B13 8JH
Cust_00003	Khasan Cai	456 Oak Avenue	Manchester	M2 4WD

▲ Figure 12.8 Customer data table; the database is in third normal form (3NF) but this table remains unchanged

Cakes table: see Figure 12.9.

Cake_ID	Cake_type	Size	Cake_cost
Birth_sml_2t	Birthday	Small	£95.00
wed_lrg_3t	Wedding	Large	£350.00
birth_lrg_2t	Birthday	Large	£150.00
birth_med_2t	Birthday	Medium	£110.00

▲ Figure 12.9 Cakes data table in third normal form (3NF)

Order table: see Figure 12.10.

Order_ID	Customer_ID	Cake_ID	Cake_description	Deposit_paid	Deposit_value	Remaining	Baker_ID
DC_000001	Cust_00001	Birth_sml_2t	2 tier, yellow frosting, "Happy 18th Birthday"	Yes	£23.75	£71.25	DCE_0001
DC_000002	Cust_00002	wed_lrg_3t	3 tier, 1x fruit, 2x sponge, Wedding Design 3c	Yes	£87.50	£262.50	DCE_0002
DC_000003	Cust_00003	birth_lrg_2t	Chocolate, Novelty Design – Car	No	£37.50	£112.50	DCE_0001
DC_000004	Cust_00003	birth_med_2t	2 tier, vanilla sponge	Yes	£27.50	£82.50	DCE_0003

▲ Figure 12.10 Order data table in third normal form (3NF)

Bakers table: see Figure 12.11.

Baker_ID	Given_name	Family_name	Salary_scale
DCE_0001	John	Simpson	A
DCE_0002	Mariami	Kaur	B
DCE_0003	Gerwas	McGregor	B

▲ Figure 12.11 Bakers data table in third normal form (3NF)

THINKING LIKE AN IT SPECIALIST

Sometimes data normalisation can be defined in four levels rather than just three:

- First normal form (1NF)
- Second normal form (2NF)
- Third normal form (3NF)
- Boyce-Codd normal form (BCNF).

Investigate how BCNF differs from 3NF and how it is used in databases.

EXAM TIP

In any normalisation question you will only be expected to normalise as far as third normal form. You will not be required to demonstrate understanding of Boyce-Codd normal form.

SKILLS

CRITICAL THINKING, PROBLEM SOLVING, ANALYSIS

ACTIVITY 5

Ramil is a food and drink vendor. He sells a range of food and drink out of a small shop in a popular tourist destination.

He currently keeps track of his food and drink inventory using a spreadsheet, but he has found that this has become unreliable.

Figure 12.12 shows a sample of the data that Ramil stores:

Date file updated	12/01/2024						
Product Name	**Category**	**Quantity in Stock**	**Cost Price**	**Selling Price**	**Quantity Sold**	**Supplier**	**Restock info**
Apple Juice	Beverages	50	1.5	2.5	12	BritGrocers Ltd.	Restock when quantity below 10 Date of last restock: 01/03/2024 Quantity purchased last restock 20
Potato Chips	Snacks	100	1	1.75	67	GreenFields Importers, BritGrocers Ltd.	Restock when quantity below 30 Date of last restock 01/12/2023 Quantity purchased last restock 150
Chicken Sandwich	Food	30	3.5	5	1	BritBite Wholesalers	
Bottled Water	Beverages	80	0.75	1.25	58	BritGrocers Ltd.	
Chocolate Bar	Snacks	75	1.25	2	100	MidlandsMunch Merchants, GardenGourmet Supplies	
Cheese Pizza	Food	20	5	7.5	15	RoyalFoods Suppliers	
Orange Soda	Beverages	60	1.25	2	23	GardenGourmet Supplies, MidlandsMunch Merchants	
Pretzels	Snacks	50	1.5	2.25	46	MidlandsMunch Merchants	
Caesar Salad	Food	25	4	6	31	GardenGourmet Supplies	New product

▶ Figure 12.12 Food and drink inventory spreadsheet

Develop a normalised database solution for Ramil.

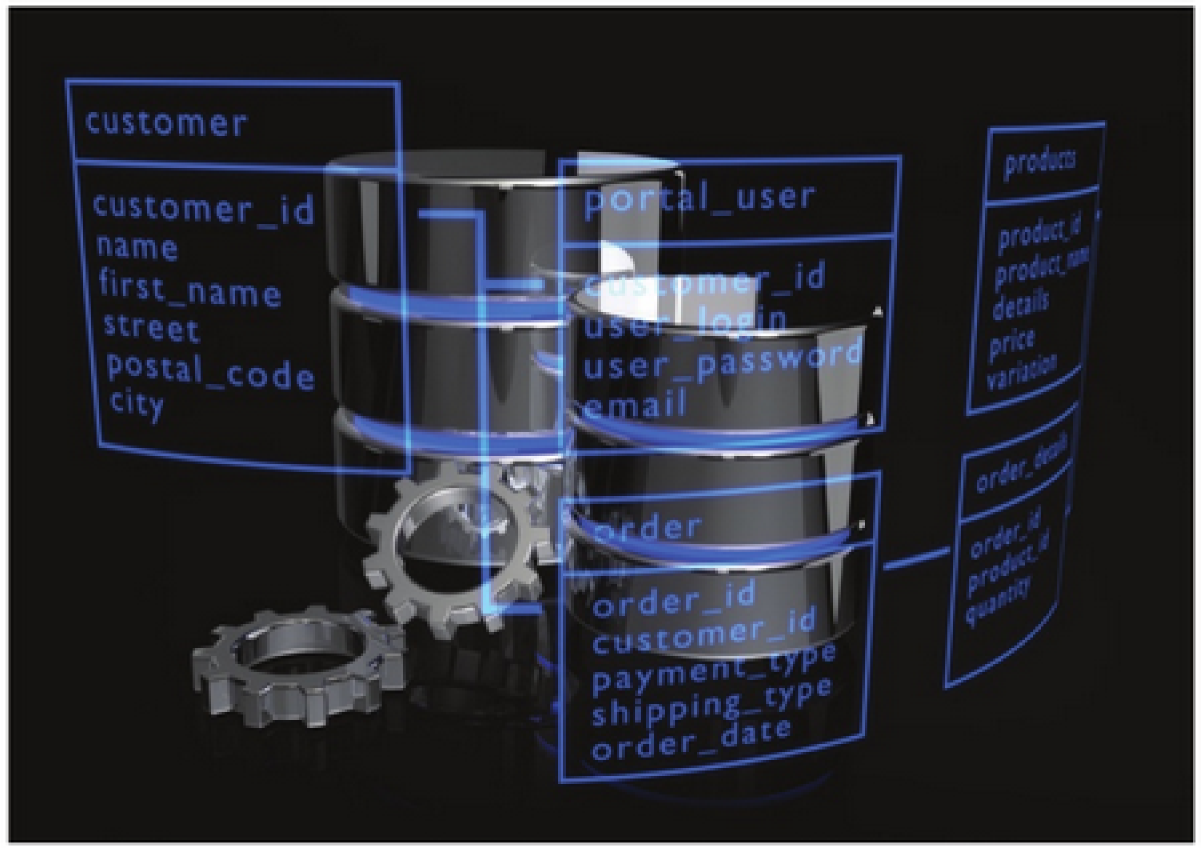
Create entities showing the database in third normal form.

KEY TERM

database management system (DBMS): a type of software application that provides tools for creation and manipulation of databases and the data that they hold

LOGICAL DATA MODELS

A logical data model is a plan for how data elements within a database are structured and relate to each other. It allows us to provide a high level view of how the data is organised focusing on the entities, attributes and relationships of the data. A logical data model is often represented in an entity-relationship diagram (ERD), which provides a visual representation of how the data is structured. When planning a database you would use the ERD along with a data dictionary to plan the database before implementing it in your chosen **database management system (DBMS)** software.



▶ Representing the data model in a graphical format makes it easier to identify relationships and reduce redundancy

There are a number of key components of a logical data model, some of which you will already be familiar with from the data dictionaries section of this chapter. These components are as described here.

Entities

These represent the real-world objects or concepts that are important to the business (e.g. customers, products, orders). Each entity is a single table in a relational database. When defining each entity it is important to ensure that the structure of the data is normalised as far as possible.

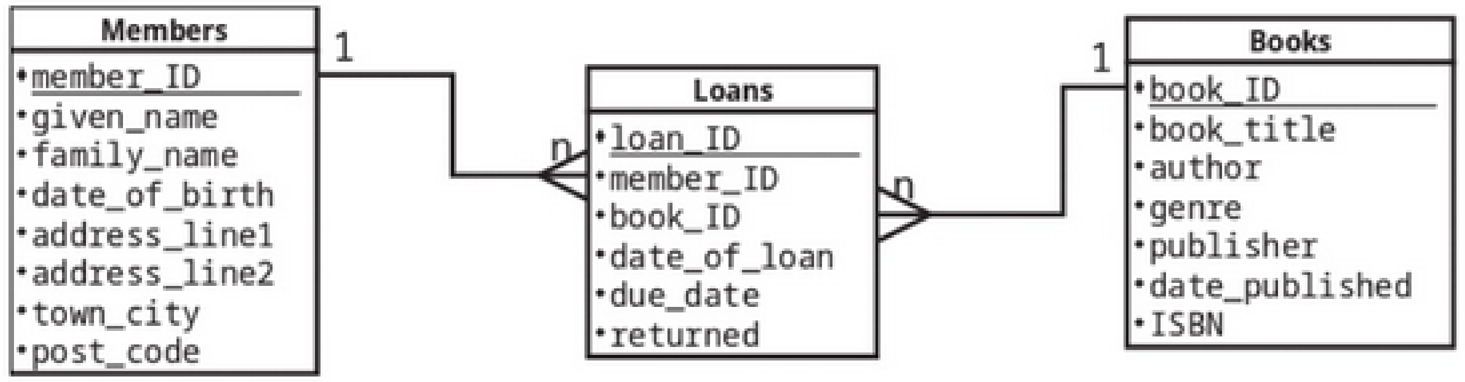
Attributes

These are the characteristics or properties that describe entities (e.g. customer name, product price, order date). These are the fields in your database. Remember that when building your data model, your attribute (field) in each entity (table) should be atomic.

Relationships

These define how entities are connected to each other and how the data and tables will interact when the database is deployed (e.g. a customer places an order, an order contains products). These are the one-to-one, one-to-many, and many-to-one relationships that you are familiar with.

The relationship type is shown by the type of line used and the use of a (1,n) notation, as shown in Figure 12.13.


Members
•member_ID
•given_name
•family_name
•date_of_birth
•address_line1
•address_line2
•town_city
•post_code
1
n
Loans
•loan_ID
•member_ID
•book_ID
•date_of_loan
•due_date
•returned
n
1
Books
•book_ID
•book_title
•author
•genre
•publisher
•date_published
•ISBN

▶ Figure 12.13 Part of an ERD for a library system; notice that the 'many' side of the relationship is denoted by a three-pronged line

Unique identifiers

Each field in a table should have a unique identifier (name) that allows it to be easily identified (e.g., customer ID, product ID, order ID). Avoid ambiguity when selecting identifiers. For example, many tables may contain some form of date field, so ensuring that an identifier is meaningful will help avoid confusion or incorrect assumption. Adding context or extra detail to the identifier is useful in these situations (e.g. order_date, delivery_date).

Constraints

Many of the constraints for the database will already be defined in the data dictionary, but these rules should be kept in mind when designing the logical data model to ensure consistency. However, the ERD should show the type of link (e.g. one-to-one, one-to-many), and clearly show the primary and foreign keys in each table that ensure data integrity and consistency (e.g. not allowing duplicate values, enforcing referential integrity).

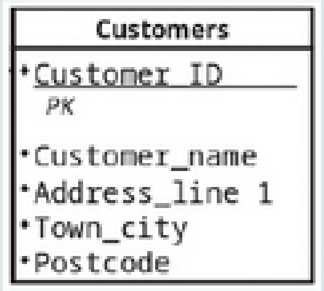
CASE STUDY: DARCEY'S CAKES – LOGICAL DATA MODELS

Darcey's bakery business specialises in making celebration cakes for weddings, birthdays and other events.

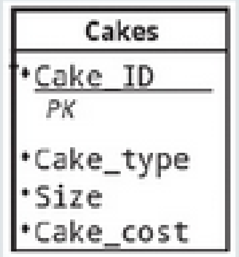
Their old system for tracking order data has now been normalised and requires a logical data model in preparation for deployment as a fully working database.

The data model will implement the version of the data in third normal form, a shown on pages 15–16.

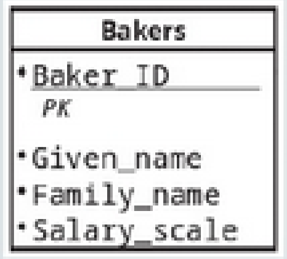
Here you can see the entity "Customers". It contains the customers' contact details. The attribute "Customer_ID" will be the primary key for this entity. In this diagram the primary key is underlined. There are many different ways to show that an attribute is a primary key including, emboldening the word, annotating with "PK", or sometimes a small key icon.


Customers
•Customer_ID
PK
•Customer_name
•Address_line 1
•Town_city
•Postcode

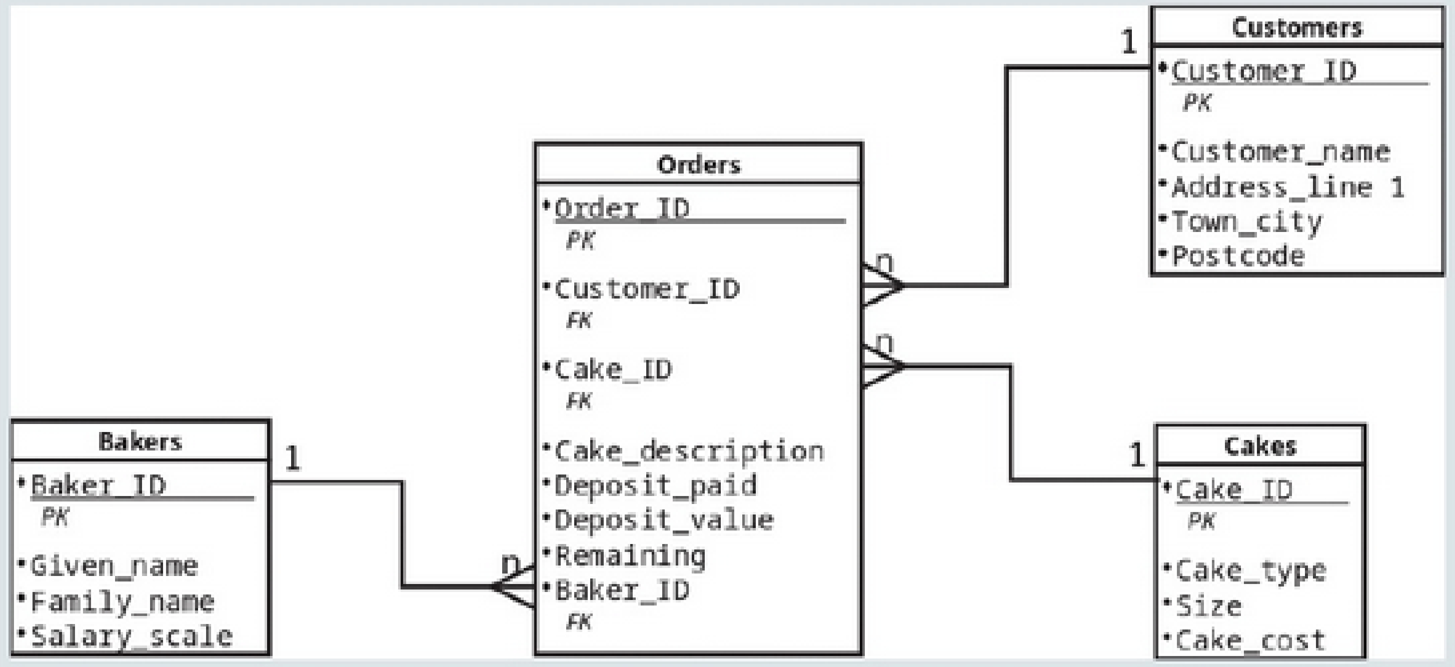
This is the entity "Cakes". It contains the details of each of the cakes that the bakery produces. The attribute "Cake_ID" will be the primary key for this entity.


Cakes
•Cake_ID
PK
•Cake_type
•Size
•Cake_cost

This is the entity "Bakers". It contains the persona information about the bakery's employees. The attribute "Baker_ID" will be the primary key for this entity.


Bakers
•Baker_ID
PK
•Given_name
•Family_name
•Salary_scale

The entity "Orders" contains details of the orders placed by customers. The attribute "Order_ID" will be the primary key for this entity. Each of the other tables are linked to this entity using their primary key. The attributes "Customer_ID", "Cake_ID" and "Baker_ID" are foreign keys in this entity.


Customers
•Customer_ID
PK
•Customer_name
•Address_line 1
•Town_city
•Postcode
1
n
Orders
•Order_ID
PK
•Customer_ID
FK
•Cake_ID
FK
•Cake_description
•Deposit_paid
•Deposit_value
•Remaining
•Baker_ID
FK
n
1
Cakes
•Cake_ID
PK
•Cake_type
•Size
•Cake_cost
Bakers
•Baker_ID
PK
•Given_name
•Family_name
•Salary_scale
1
n

SKILLS
REFLECTION

ACTIVITY 6

A small hotel wishes to improve the way that it stores and manages data. The hotel staff currently use a series of spreadsheets to store their data, causing significant data redundancy that has resulted in a number of issues.

1. Produce a data dictionary and logical data model (ERD) to plan a normalised data solution for the hotel.

   The data solution should effectively:

   - store information about which rooms are booked and which are available at any given time
   - keep track of the cleaning of rooms
   - store details of customers/guests and staff.

   The hotel's owner has provided you with this information to help you:

   - The hotel has 25 rooms.

- Types of rooms available:
    - Double
    - Single
    - Twin
    - Family.
- Room cleaning:
    - Rooms are cleaned every other day if guests are staying more than one night.
    - Rooms are cleaned again once the guests check out.
    - A room cannot be assigned to a new guest until it has been cleaned.
- The hotel employs three types of staff:
    - Reception
    - Cleaning
    - Kitchen.

You should include any entities and attributes that would effectively meet the hotel's needs.

2. Discuss your data dictionary and ERD with a partner. Use these prompts to help your discussion.

- Why have you organised the data in the way that you have?
- What features of your data design are effective?
- What additional entities and attributes could you add to the structure to better meet the hotel's needs?
- What additional features could be added to the database system to make it even more useful to the hotel owner?

BIG DATA

Over recent years the use of computer systems has become integrated into almost all aspects of our lives. This means that data is being constantly generated and the amount of data generated and used continues to grow. Big Data refers to the very large and complex data sets that are generated by these computer systems. Because of their size and complexity these data sets are very difficult to manage using traditional data-processing tools. However, this data can provide vital insights for organisations into many different areas of their operations. As such, utilising Big Data is a priority for many organisations.

THE FIVE Vs

There are five key concepts when dealing with Big Data, which are referred to as the 'five Vs': volume, velocity, variety, veracity and value. Understanding these concepts allows us to better understand the characteristics and challenges of collecting, storing and processing data. It allows us to use the appropriate tools and approaches so that we can effectively use the data to gain valuable information.

Volume

Volume refers to the amount of data. As already mentioned, Big Data involves massive amounts of data, which continue to grow. Many systems generate data constantly. Sources such as social media and business transactions all generate data. The volume of data has massive implications for organisations in terms of storage and processing requirements.

Velocity

Velocity is the speed at which data is generated. Big Data is often generated at exceptionally high speeds. Social media, for example, generates massive amounts of data at a very high rate. For example, in December 2023, Facebook generated 4 000 000 'likes' and 360 000 X (Twitter) posts were sent per minute (*source: Statista 2024*).

Variety

In the data integrity section, you learned about ensuring data is consistent in terms of structure and format. One major challenge of dealing with Big Data is that you are often pulling data from many different sources. The issue is that because of the range of sources and the different ways that the data is generated, Big Data might often come in a wide range of formats, such as:

Structured data: well-organised data, which is much easier to work with and analyse. This includes traditional DBMS style organisation.

Unstructured data: data that is not organised using typical data tables and logical data models. This data is often collections of files that include text, images and videos, which are very difficult to manipulate and analyse.

Semi-structured data: although not structured in a traditional relational database fashion, semi-structured data does contain some elements that make it easier to manipulate than unstructured data. Semi-structured data will typically contain some metadata that can be used to help with processing.

SKILLS
CONTINUOUS LEARNING

ACTIVITY 7

Use the internet to do some further research on structured, unstructured and semi-structured data, including how it is generated and the types of data files each uses.

Produce a short presentation to summarise your findings.

Veracity

Veracity refers to the quality of the data within the data set. Owing to the variety of sources that the data is generated by and extracted from, there can be uncertainty as to the accuracy of the data. For example, data extracted from social media could be biased or contain misleading information, because of its user-generated and unregulated nature. It is a significant challenge to ensure that any potential variability in quality and accuracy is considered and mitigated when working with Big Data.

Value

There are significant challenges to working with Big Data, however there are also significant rewards. Successfully utilising big data can bring significant value to a company or organisation. For example, gaining detailed insights into consumer trends can lead to more effective strategic decisions, or identifying where products and services can be improved helps to increase customer satisfaction and attracts new customers.

INFRASTRUCTURE AND SERVICES FOR BIG DATA

WWW
BIG DATA

▶ Big Data is often generated and used by the IoT (Internet of Things)

KEY TERM

infrastructure: hardware and software resources that support data and processes

As you will know from the previous section, Big Data involves gathering data from a wide range of sources that relies on a vast array of **infrastructure** and services. These can be broadly categorised into three main groups:

- Collection
- Storage
- Transmission.

Collection

The data in these large data sets is generated by a wide range of devices and services. These sources include those listed.

KEY TERM

IoT (internet of things): interconnection of everyday objects through the internet to allow the sending and receiving of data

- Social media: platforms like X (formerly Twitter), Facebook, and Instagram generate vast amounts of data through user interactions, posts and updates.
- **IoT** devices: sensors and connected devices, such as smart appliances, wearables and industrial sensors, continuously generate data. For example, you may have a smartwatch or fitness tracker that can count the number of steps you take each day, monitor your heart rate or even track your location.
- Business transactions: e-commerce, financial transactions and other business-related activities generate large data sets. For example, every time you make a purchase or use a service, you generate data, the data generated by that interaction can be collected and analysed.

- Web and app logs: data generated by users interacting with websites, applications and online services. Web servers keep track of every visit to a website, which can be used to analyse user behaviour such as how long the user spends on a site or if they are a returning visitor.
- Surveillance systems: video footage, images and other data collected for security and surveillance purposes.
- Machine-generated data: as process automation increases, so does the amount of data produced by these systems. Machine-generated data can vary significantly in its scope but common types of machine-generated data include weather and climate monitoring systems data, as well as data from manufacturing processes.
- Publicly available data: government databases, research data sets and other publicly accessible sources contribute to Big Data.

Storage

The storage of the ever-increasing data sets is a significant challenge for companies and individuals wishing to make use of Big Data. The cost of purchasing and maintaining storage space, and the need for storage capacity to be scalable as needed, can be prohibitive. Many companies no longer use their own 'in house' resources, preferring instead to make use of cloud services, which allows them to pay for storage and systems on an 'as needed' basis.

Many of the large companies that offer cloud storage and services for Big Data offer tools that work alongside their storage to allow users to manipulate and analyse Big Data sets. Popular cloud services include:

- Amazon web services (AWS)
- Microsoft Azure
- Google Cloud Platform (GCP).

As well as the storage and processing services provided by the cloud providers, Big Data needs storage systems to hold the data so that it can be manipulated. These systems sit on top of the cloud infrastructure and can often be used on any cloud service allowing flexibility and portability. Common storage systems include the following:

- Distributed file systems: these allow the storage of large volumes of data across multiple nodes. This allows the system to be easily scalable and improves fault tolerance.
- NoSQL databases: used for handling unstructured or semi-structured data, providing flexibility and scalability.
- Data warehouses: a central storage point where data from a range of sources is brought together. These platforms provide scalable and high performance storage for structured and semi-structured data, which is optimised for analytics.

Transmission

Transmission refers to the moving or transfer of data in and between systems. Big Data transmission presents some interesting challenges owing to the sheer size, complexity and rapidly changing nature of the data. These challenges are explored in more detail later in this section; however one of the key issues is how to transfer large data sets in such a way that the data quality is retained and the very nature of accessing the data will not cause issues for the user. A range of infrastructure solutions can be used to support data transmission. These include the following:

KEY TERMS

bandwidth: the amount of data that can be transmitted over a connection in a given amount of time

silo: a collection of data used and controlled by one group, department or business that is not accessible by the rest of the organisation

High-bandwidth networks: fibre optic cables and dedicated network connections offer high **bandwidth** to accommodate large data volumes.

Distributed storage: distributing data across multiple storage nodes reduces network strain and improves accessibility.

Data compression: compressing data before transmission reduces file size and bandwidth requirements.

Cloud-based solutions: storing and accessing data on a cloud storage platform reduces the amount of data that needs to be downloaded to the user's computer.

IMPACT OF STORING BIG DATA

The massive size of the data sets presents specific challenges for organisations when storing this data. Users need to be able to access and use the data easily, and organisations have a responsibility to ensure that the data is protected.

Access

There are a number of factors relating to data access, which can vary depending on the point in the business and analysis process that the action occurs.

Data silos

Big Data often relies on gathering data from different data **silos** stored across different departments or systems. Accessing this data so that it can be integrated from these different sources can be complex, requiring effective data integration strategies to enable data to be analysed.

On-demand access

Those that use the data need to be able to use the data whenever and wherever they need. Organisations must put systems in place that allow reliable access to data. Many organisations will rely on cloud services to provide storage systems that will have limited downtime and reliable data backup.

Cross-border data access

Large companies often have offices and workers all around the world. Accessing and sharing big data across international borders may be subject to various legal and regulatory challenges. Organisations need to navigate complex regulatory environments to ensure compliance.

Processing time

Once a user has access to the data, they need to be able to use the data effectively. Ensuring that data is processed in a timely fashion is important to maintain productivity.

KEY TERM

latency: the delay between requesting and receiving data

Real-time access

Some applications and analytics require real-time access to Big Data. Achieving low-**latency** access can be challenging, and organisations may need to invest in technologies that support real-time data processing. This may include local resources such as high-powered computers and servers, or often also cloud services provided by third-party vendors.

Processing speed

Analysing large data sets can involve hundreds and thousands of calculations by a computer system; even with modern high-speed processors and servers there is still a limit to how fast these calculations can be processed. If a user requires 'real-time' data often a system will be designed so that the needs of the user are predicted and much of the data processing is done ahead of time. Fewer processes then need to be carried out when the user wants to access the data, which would reduce the perceived processing time for the user.

Transmission time

As you learned in the infrastructure section, there are a number of challenges in ensuring that data is transmitted in and between systems in a timely manner. It is important that any system minimises the latency experienced by the user particularly in real-time applications, or in time-critical systems.

Security

As the volume of data increases, so does the importance of securing it. Data security is often thought of in two main components: data privacy and data security. Maintaining data privacy and data security is a key requirement for organisations, both in terms of ensuring appropriate legal and regulatory guidelines are followed, and also in ensuring customer confidence is maintained.

Data privacy

Big Data often contains sensitive information. Ensuring compliance with data protection regulations and implementing robust privacy measures are crucial to prevent unauthorised access and to protect the privacy rights of individuals.

Data security

With the increasing volume of data, security becomes a significant concern. Implementing strong access controls, encryption and other security measures is essential to safeguard against data breaches and cyber threats.

SKILLS
CONTINUOUS LEARNING

ACTIVITY 8

Investigate the tools and features provided by cloud services to ensure data privacy and security.

In your investigation you should consider:

- how data security and privacy measures are implemented
- the processes involved in establishing and maintaining an effective information security management system (ISMS)
- the benefits and drawbacks for an organisation of implementing privacy and security measures.

Make detailed notes on your findings from your investigation.

DATA MINING, WAREHOUSING AND ANALYTICS

Data mining, data warehousing and data analytics are three closely related concepts that have important roles in managing, processing and deriving insights from Big Data.

Data mining

Data mining is the process of extracting meaningful information from the data. It uses statistical techniques such as clustering, classification, regression and anomaly detection. Organisations can extract meaning and insight that may seem initially hidden in the data. They often use machine learning and artificial intelligence to assist with this process.

Data warehousing

Data warehousing involves the collection, storage and management of the raw data from a range of different sources. Data warehousing will combine the data from an organisation's own data silos with external data into a single data repository. A data warehouse is a centralised repository that provides a unified view of the data, making it easier for decision-makers to access and analyse information.

Data analytics

Data analytics takes the information derived from data mining and turns it into actionable intelligence for an organisation. It involves the use of data analysis and statistical methods to gain insights from the information, make predictions and inform the organisation's decision-making process. Data analytics uses a variety of tools. Some of the most common functions include:

- producing reports, dashboards, charts and graphs
- natural language search, text analysis
- machine learning
- statistical analysis
- modeling and simulations
- data import and export to other software
- data security.

There are three main strands to data analytics:

Descriptive analytics is the process of summarising historical data to gain insights into what has happened. This is used alongside diagnostic analytics to gain an understanding of why certain events occurred, which can help replicate or improve good practice or identify errors so that they can be avoided in the future.

Predictive analytics looks at a range of factors to identify causes and effects and make predictions about future outcomes. By analysing historical data it is possible to predict what might happen if the same issues reoccur. For example, by analysing weather patterns, it may be possible to predict that an increase in temperatures and lack of rainfall in a particular region will reduce the amount of a particular food type available, which in turn would increase its price.

Prescriptive analytics is often linked to predictive analytics. Once a trend or possible outcome has been identified, prescriptive analytics suggests the actions required now to optimise results. Consider, for example, a large food manufacturing company. Predictive analytics may suggest that a change in weather patterns could result in the shortage of a food type that the company relies on as a main ingredient. It may suggest different ways to minimise the impact of this shortage. For example, it may predict buying more of the ingredient before the shortage to ensure that the organisation has spares, or it may identify different suppliers from different regions, or even make predictions on how to minimise the impact of the cost increase.

THE USE OF BIG DATA BY INDIVIDUALS, ORGANISATIONS AND SOCIETY

As organisations see the benefits of Big Data, it is becoming increasingly integrated into all areas of society. From personalised advertisements and recommended purchases on e-commerce sites to detecting issues in financial transactions, Big Data provides valuable information and influences many areas of our lives.

Healthcare

The ability to analyse large quantities of data and use computer systems to discover patterns and anomalies has transformed healthcare in recent years. Big Data continues to have significant impact on doctors, patients and the treatments and technologies within the field. Common uses of Big Data include the following.

Personalised treatment plans

Analysing patient data and comparing it with markers identified in thousands of other data points allows medical professionals to more accurately predict disease risks or how a patient might respond to certain treatments. This allows for more tailored treatment options that can improve patient outcomes.

Accelerate drug discovery and development

By analysing large data sets that contain genetic and clinical data, doctors can identify trends in which drugs and treatments proved most effective in different cases, allowing them to target specific treatments or combine approaches to develop new treatments.

Track and manage public health threats

Healthcare professionals and government agencies have monitored and analysed data related to disease outbreaks and risks factors for years. The utilisation of Big Data and the associated tools and services allows them to draw inferences from a much wider pool of data and make more accurate analyses. This analysis can be used to inform public health interventions.

Infrastructure planning

Town and city infrastructure involves a wide range of resources and services. From roads and public transport to management of utilities such as electricity supplies, the use of Big Data helps improve the quality and efficiency of services.

Predictive maintenance

Big Data analytics can be used to predict and prevent infrastructure failures. By analysing data from sensors, monitoring equipment and even historical data, predictions can be made for when maintenance and replacement of vital systems and components will be required. This enables proactive maintenance allowing actions to be taken before serious issues occur, therefore reducing downtime and extending the lifespan of critical infrastructure components.

Traffic management and urban planning

Big Data is employed to analyse traffic patterns, collect real-time data from sensors and GPS devices, and optimise transportation networks. This information helps urban planners make informed decisions about road design, public transportation routes and traffic signal timings.

Utility infrastructure optimisation

Big Data analytics helps in optimising the efficiency and quality of the infrastructure used to deliver utilities to homes and businesses. By analysing data from sensors and related systems, companies can look at usage patterns, consumption levels, system performance and even weather forecasts to make predictions as to when and where energy and water demand will fluctuate. This assists in optimising distribution, storage and maintenance of systems, as well as planning for further development and improvement of systems.

Transportation

In addition to ensuring road planning is optimised to ensure roads and public transportation will be efficient, Big Data is used in a number of ways to improve transportation.

Public transportation optimisation

Big Data can be applied to optimise public transportation systems. By analysing passenger data, monitoring traffic patterns and route efficiency, routes and schedules can be adjusted based on demand patterns. This can lead to improved service reliability and better alignment with passenger needs, which can also increase public transport usage.

Fuel efficiency and emission reduction

Big Data analytics is used to monitor fuel consumption, vehicle performance and emissions. This data helps fleet operators optimise fuel efficiency, reduce carbon emissions and comply with environmental regulations.

Safety management

The analysis of Big Data analytics can have a meaningful impact on the safety of roads and transportation. Data sets relating to accidents, near misses and other safety-related incidents helps identify common risk factors and implement preventative measures. These measures can then be continually monitored to review their effectiveness and refine solutions as required.

Fraud detection

Big Data has proven to be effective in helping detect fraud by enabling organisations to analyse vast and diverse data sets that would previously have been too disconnected to easily interrogate. Using Big Data allows organisations, such as banks, insurance companies and even law enforcement organisations to identify patterns, anomalies and suspicious activities that may indicate fraudulent behaviour.

Pattern recognition

Big Data analytics helps identify patterns of normal behaviour within large datasets. Deviations from these patterns can signal potential fraud. Machine learning algorithms can be trained to recognise these patterns and detect anomalies that may indicate fraudulent activities.

Behaviour analysis

Analysing user behaviour and transaction patterns can help identify unusual or suspicious activities. Big Data analytics can track and analyse historical data to establish a baseline for normal behaviour and detect deviations that may indicate fraudulent behaviour. For example, banks may use this to track when and how purchases are made from an account. If a purchase is made that does not fit usual patterns (e.g. much more expensive than normal, in a different country) then this may suggest fraudulent activity and the bank will contact the account holder to check.

Real-time monitoring

Big Data enables real-time monitoring of transactions and activities. By processing and analysing data in real time, organisations can quickly identify and respond to potential fraudulent activities, minimising financial losses for both the organisation and its customers.

THINKING LIKE AN IT SPECIALIST

As the use of Big Data increases in a number of areas, such as healthcare, infrastructure planning and fraud analysis, so do the concerns relating to the use and storage of data.

What concerns do you think the use of Big Data may cause for:

- organisations
- customers/clients
- the general public?

Investigate how these concerns can be mitigated.

CHECKPOINT

1. Explain the importance of data integrity and the methods of maintaining it (data governance).
2. What is a data dictionary and how is it used to plan a database?
3. Why is it important to use data validation in a database?
4. What is data redundancy and what problems does it cause?
5. Describe the purpose of normalisation.
6. What is a logical data model and what is it used for?
7. Explain what is meant by Big Data.
8. Define the five Vs of Big Data.

EXAM TIP

When responding to a question with the 'explain' command word you should make sure that you:

- state an initial point
- add an expansion that demonstrates deeper understanding of the point you have made.

The number of points/expansions that you make will depend on the mark tariff for the question.

For example, a question that is 'Explain one reason why...' with a mark tariff of three marks, will require an identification, an expansion and then a further expansion or example. The second expansion/example should build upon the first expansion.

SKILLS

CRITICAL THINKING, PROBLEM SOLVING

EXAM PRACTICE

1. Velocity is one characteristic of Big Data.

   Give **two** other characteristics of Big Data. [2 marks]

2. Explain one infrastructure requirement for Big Data. [2 marks]
3. Describe how Big Data can be used in the healthcare sector. [3 marks]
4. Tyshawn runs a clothing store.

   All data for the store is currently stored in a spreadsheet.

Product name	Company name	Company telephone	Company email	Order time	Sizes in stock
Holo Hoodie	VortexVogue	7700 900430	info@vortexvogue.com	1 day	M, L
Quantum T-Shirt				7 days	S, M, L
CyberPulse Gloves				3 days	L, XL
Denim Jacket	SolarFlair	01632 960072	sales@sflair.org	2 days	S, M, L, XL
Quantum T-Shirt				1 day	S, M
Nebula Skirt				4 days	S
Solar Shirt				2 days	M, L
RetroActive Jeans	RetroFusion	7700 900147	retro@retrofusion.com	5 days	L
Galactic Sunglasses				4 days	N/A

   Tyshawn wants to replace the spreadsheet with a database.

   (a) Develop a normalised database solution to replace the spreadsheet.

   (b) Create entities showing the database in third normal form.

   (c) Identify all primary keys and foreign keys. [9 marks]

TOPIC 13 ENABLING TECHNOLOGIES

The way in which we use and interact with computer systems is a key factor in whether or not that system will meet user needs. As technologies have advanced, they have allowed us to adopt approaches to problems that mean users can access tools and environments that otherwise would not be feasible. In this topic, you will explore how computer systems can be used to provide access to alternative systems, how they can be used to share and protect data, and different ways that users can interact with these systems.

TOPIC 13 ENABLING TECHNOLOGIES

LEARNING OBJECTIVES

By the end of this topic you should be able to understand:

- what virtualisation is and why it is used
- ways that virtualisation can be achieved
- how distributed systems are used to share data
- the issues associated with distributed systems
- the importance of human computer interaction
- ways that human computer interaction can be implemented
- the ergonomic principles associated with human computer interaction
- how interfaces are designed to meet the needs and user requirements
- how data is stored in the cloud
- how data is secured in cloud-based systems
- the function of database management systems (DBMS) in controlling access to data
- the features and functions of symmetric encryption
- the features and functions of asymmetric encryption
- the features and functions of certificate-based encryption.

GETTING STARTED

Think about a time when you have used an online version of an office application such as Microsoft Word. Compare the online version to a version that is installed on a computer. What differences do you notice between the two versions? What do you notice that is the same?

File
Edit
New
Ctrl+N
Open
Ctrl+O
Save
Ctrl+S
Save As...

▲ Online software may provide fewer features. Why do you think that is?

VIRTUALISATION

It may not always be possible to have access to a specific piece of software or hardware. Consider, for example, a computer games developer. They may be developing a game that will be used on a range of consoles and personal computers; however, it would not be viable for them to make the game directly in all the systems. Virtualisation, however, would allow the developer to create an environment that looks and behaves like the target system – allowing them to test the game in that environment. In this section, you will explore more deeply why virtualisation is used and how it can be achieved.

WHAT IS VIRTUALISATION?

Virtualisation is the act of creating a virtual version of something, such as a computer system, server, storage device or network resource. The virtual version can then be used in a similar way to the original.

There are different types of virtualisation, including desktop, storage and network virtualisation.

Desktop virtualisation

This type of virtualisation allows you to run multiple operating systems on a single desktop computer. For example, it means being able to run a Microsoft Windows environment on an Apple computer. **Thin clients** are another example of desktop virtualisation.

KEY TERMS

ecosystem: a set of tools, libraries and applications designed for a particular purpose or platform e.g. Apple, Android

thin client: a computer that runs resources stored on a remote server rather than locally installed resources

Storage virtualisation

This type of virtualisation allows you to pool together multiple physical storage devices into a single virtual storage pool. That is to say, a system could have many different storage devices connected, but these will appear as a single virtual storage system, which simplifies management by allowing them to be managed as one system.

Network virtualisation

This type of virtualisation allows you to create virtual networks that are independent of the underlying physical network infrastructure. This allows the design of a network to be more flexible, and enables parts of the network to be isolated. For example, a school could separate the network resources that are accessible to staff, such as student data and administrative systems, from the resources used for teaching and learning.

SKILLS

INTELLECTUAL INTEREST AND CURIOSITY

ACTIVITY 1

The Google Chromebook is an example of a modern implementation of a thin client.

Use the internet to find out more about how the Google Chromebook provides access to tools and resources.

BENEFITS OF VIRTUALISATION

Whether it is improving ease of access to particular software or reducing costs, effective use of virtualisation can have a positive impact on workflow and provides a number of benefits for both individuals and organisations.

Resource utilisation

Virtualisation maximises the use of already owned physical resources. Being able to make use of tools and services from different **ecosystems** on a single platform can improve efficiency and provide cost savings.

Isolation

Virtualisation provides separation between the virtual system and the platform it is being used on, and from other virtual instances. Therefore, any issue that occurs within the virtual instance, for example malware, is restricted to only that virtual instance and is prevented from affecting other parts of the same physical system.

Flexibility and scalability

Virtualised environments are more flexible and scalable, making it easier to adapt to changing workloads and requirements.

KEY TERM

snapshot: a saved copy of a system, including all data and settings, at a given point in time

Disaster recovery

Virtualisation facilitates the creation of **snapshots** and backups, making it easier to recover from hardware failures or other disasters. Whereas backups tend to focus on the data of a system, a snapshot allows the user to make a copy of an entire system, including software and settings. In the event of a system failure the entire snapshot can be used to restore the system in the exact state it was at the time of the snapshot.

DRAWBACKS OF VIRTUALISATION

Although virtualisation provides many benefits, there are, as with many systems, some drawbacks that need to be considered before selecting a virtualised solution.

Complexity

Setting up and managing a virtual environment can be complex. The process of setting up the virtual environment requires specialist software and skilled personnel. For example, providing access to the internet inside the virtual environment, while still maintaining isolation and security, requires more steps than it on the host environment.

Cost

While virtualisation can lead to cost savings in terms of reducing the variety of systems that need to be purchased, the initial setup can be expensive. The cost of the virtualisation software and powerful hardware that is capable of running all the virtual instances can be high. In addition, a company may need to provide additional training for its IT staff.

Performance

Virtual solutions share the resources of the underlying physical hardware. There is also often an additional level of software between the hardware and virtual solution. This can lead to performance bottlenecks if too many virtual machines (VMs) are running, or if the physical hardware is not powerful enough, or if the virtualisation software is incorrectly configured.

THINKING LIKE AN IT SPECIALIST

It is important to consider the benefits and drawbacks of any IT system before selecting a solution. What may work best in one situation may not be suitable in another. Like any IT solution, virtualisation is a tool that should be used selectively, and only when appropriate.

ACHIEVING VIRTUALISATION

There are two main approaches to virtualisation: containerisation and virtual machines. Virtual machines, as the name suggests, simulate an entire computer system. When using a virtual machine, the software will typically provide access to a complete operating system, allowing the user to interact with the system as if they are using that system directly. Containerisation uses a much narrower focus and provides access to a specific function or application only.

Table 13.1 shows a comparison between virtual machines and containerisation.

TABLE 13.1: COMPARISON OF CONTAINERISATION AND VIRTUAL MACHINES

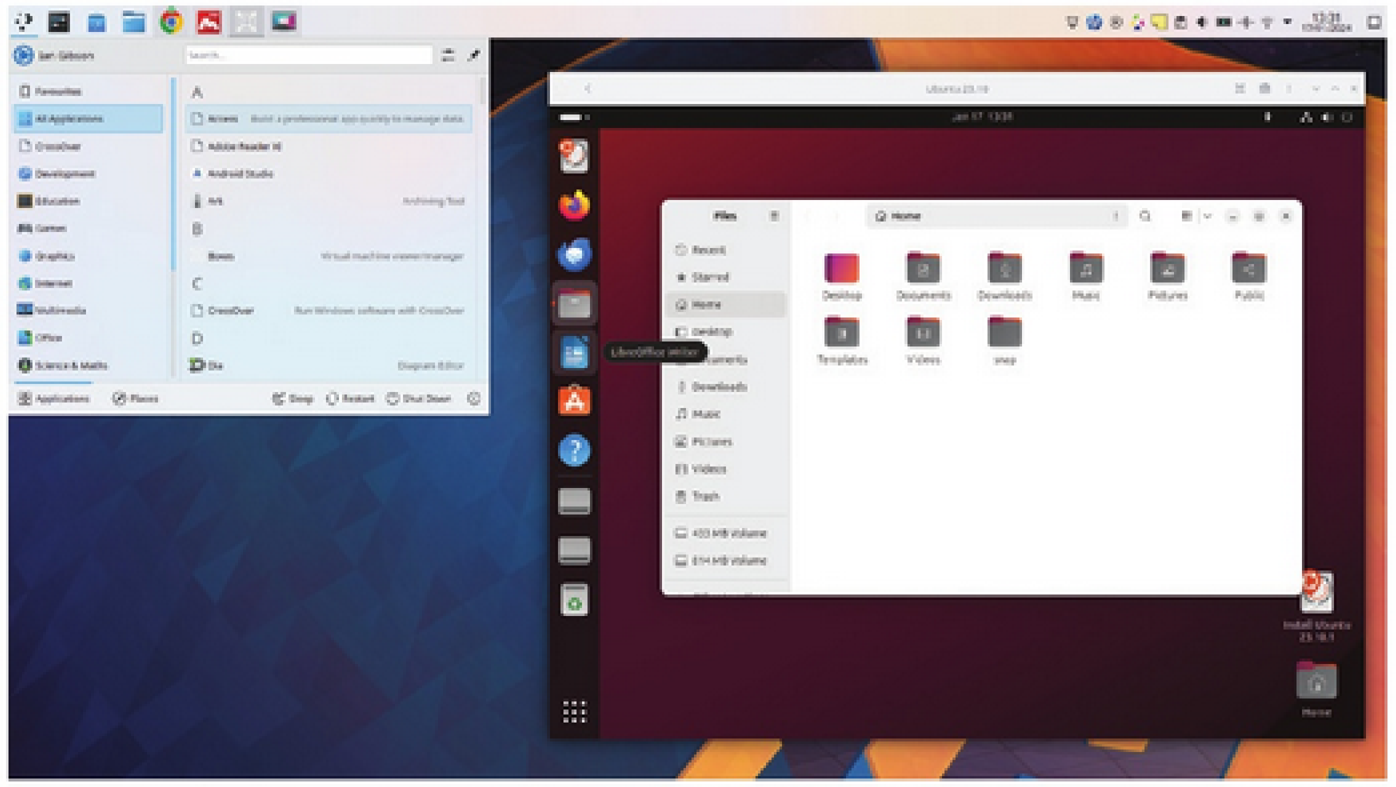
Issue	Virtual machine (VM)	Containerisation
Abstraction level	• Full hardware virtualisation • Includes a complete operating system (OS) instance • **Hypervisor** emulates the underlying hardware • Can run multiple VMs with different Operating Systems on a single physical machine.	• Operates at the application layer • Contains the application and any dependencies • Shares the host OS kernel but runs in isolated user spaces • Smaller footprint as only required dependencies included instead of full OS.
Resource usage	• Consumes more resources (memory, disc space and CPU) compared to containers • Requires more time to start and have larger image sizes.	• Minimal overhead compared to VMs • Faster startup times, reduced resource consumption and smaller image sizes.
Isolation	• Strong isolation: each VM has its own OS instance and runs on a hypervisor • Suitable for running applications with different OS requirements or security constraints.	• Lower level of isolation compared to VMs • Isolated from each other, but may share certain resources • Higher risk of security vulnerabilities if not configured properly.
Portability	• Typically less portable because they **encapsulate** an entire OS • Moving VMs between different hypervisors or cloud platforms might require additional configuration.	• Highly portable • Self-contained: they encapsulate the application and its dependencies • Easier to run the same containerised application consistently across various environments.
Uses	• Best used for running applications with wide ranging OS requirements • Strong support for legacy applications • Good for isolation/security critical situations.	• Useful for microservice architectures • Good for cloud-native applications, where rapid deployment, scalability and **portability** are key considerations.

KEY TERMS

encapsulate: to group or bundle together related data and/or functions

hypervisor: a program that runs and manages virtual machines

portability: the ease with which computing services can be transferred between systems and providers



▶ An example of a virtual machine providing access to a second operating system

SKILLS

COLLABORATION, PROBLEM SOLVING, ANALYSIS

ACTIVITY 2

A financial planning and investment company is planning on changing the way in which it organises its work force.

Currently all staff work in a large city centre office, but the company's management team has decided to move to smaller office premises and allow staff to work remotely if they wish and only visit the office if required, e.g. meeting and working with clients, professional development events.

With a partner, discuss and plan how the company could make use of virtualisation to support its IT needs.

You should consider the types of tasks that would be needed both in and outside of the office and consider how VMs and containerisation could be used to solve each of these problems.

EXAM TIP

If a question includes a scenario like the one in Activity 2, start by breaking down the issue into smaller subproblems that need to be solved.

KEY TERM

node: a key point in a network relating to a specific connection or device

DISTRIBUTED SYSTEMS

A distributed system is a collection of independent computing **nodes** that work together to perform a common goal. Although distributed systems will often contain virtual nodes, it is common to also distribute the nodes over different locations and hardware platforms. The goal of a distributed system is to provide high levels of performance, scalability and fault tolerance.

You may already be aware of the concept of distributed file systems and storage from Topic 12; distributed systems are the platforms that support and provide access to file systems and data storage.

Because of the many benefits they provide, distributed systems are becoming increasingly popular with large organisations as a way of providing access to their data and systems for staff. However, they are more commonly used for these tasks:

Web services: together with cloud computing platforms and content delivery networks (CDNs), web services are examples of distributed systems that enable users to access resources globally. Spreading content and systems to different geographical regions provides many benefits; for example, video streaming services have data and content distributed, so that even though the service is based in a different country from the user, when accessing content, it is accessed on a server within the user's country, reducing response time and improving user experience.

Data services: distributed databases and distributed file systems are common tools used by cloud services for managing large volumes of data. These services distribute and mirror data across multiple sites and systems to protect against data loss and improve access for users.

Blockchain is a more recent development and is an application used in distributed systems such as those involved in cryptocurrencies. With blockchain, transaction records (a ledger) are not stored in a single location, which reduces the risk of a single point of failure, or a single point where a transaction record can be tampered with.

THINKING LIKE AN IT SPECIALIST

The rise and fall of popularity in technologies such as blockchain and cryptocurrency are examples of how approaches in the IT industry can fluctuate. It is important to keep up to date with changes and developments in technology. Consider subscribing to mailing lists and news outlets that specialise in science and technology news.

ISSUES RELATING TO DISTRIBUTED SYSTEMS

Distributed systems are extremely complex to set up and manage, so any organisation that adopts them requires specialists to ensure the systems are set up and maintained effectively. This takes significant investment in terms of both time and money. Because of the need for significant amounts of computer hardware, and the need for this hardware to be geographically isolated from other hardware, many companies use cloud service providers to support their systems. Services such as Google Cloud Platform (GCP) and Amazon Web Services (AWS) allow companies to make use of their servers around the world to access the benefits of distributed systems.

Despite there being some challenges, distributed systems provide a range of benefits for an organisation, including those outlined here.

Failure resistance

Distributed systems are, by design, highly fault tolerant. Data and services are spread out and duplicated across many different physical systems, so there is no single point of failure. If one system malfunctions, is damaged or succumbs to a cyber attack, the system can continue to operate using data from other locations. The other systems can then be used to restore the affected system to fully working order.

Concurrency

Distributed systems often involve concurrent processing, where multiple tasks or operations occur simultaneously on different nodes. In a large organisation, this means tasks can be allocated to whichever node has capacity at any given time. This concurrency is essential for achieving high performance and responsiveness.

Replication

As you have already discovered, data is often replicated across nodes to enhance availability and fault tolerance. However, this does introduce some additional challenges. Systems must be established to ensure that data is consistent between the different instances. For example, if a transaction was updated on one system but not the others, this could lead to errors in other business areas.

Performance

Spreading data and workload between systems helps improve overall system performance. Requests can be allocated to whichever node currently has the lowest workload, thus reducing lag and increasing workflow. However, a key issue to consider is the location of the node and its proximity to the user. As distributed systems can be spread out all over the world, the system needs to be able to decide effectively if the increased transmission time of sending the data to and from a more remote location is quicker or slower for the user than sending the request to a busier but more local node.

SKILLS

REASONING, DECISION-MAKING

ACTIVITY 3

A company is providing access to distributed systems for its employees. All the employees work remotely.

How will using distributed systems help the workers and benefit the company?

In addition to the issues listed in this section, what other factors might affect the performance of the system, and how might this affect workers?

HUMAN COMPUTER INTERACTION

Human computer interaction (HCI) is an area of information technology that focuses on how users control and interact with computer systems. The aim of HCI is to allow the user to easily and effectively complete tasks in order to meet their needs.

IMPACT ON USER EXPERIENCE

User experience (UX) refers to the extent to which users find computer systems enjoyable, easy and intuitive, resulting in high levels of user satisfaction. The user interface (UI) is one of the key parts of this experience in terms of how engaging it is and how it helps the user to complete their tasks effectively. However, an appealing UI alone does not guarantee a good user experience; if a system looks appealing at first glance but is confusing to use or provides poor information, then the user experience suffers.

Usability

It is important to ensure that a user can interact with the product and achieve their goals with ease. Usability can also be influenced by many factors such as speed/responsiveness, simplicity of processes, number of steps to achieve a goal, clarity of the user interface and appropriateness of the controls.

Consistency

Consistency in design ensures that similar elements or actions are presented in the same way across different parts of the system. This helps the user to predict what will happen in different contexts, which speeds up how quickly a user will become familiar and comfortable with a system. It is also advisable to follow common conventions that have been established in other similar products, for example using recognisable and accepted icons for certain tasks such as 'Save' or 'Print'.

Feedback and communication

Providing feedback to users is crucial for maintaining a clear and meaningful communication channel between the system and the user. Feedback can be in the form of visual cues, sound or **haptic** feedback. It is important to ensure that a user understands the outcome of their actions, for example, providing feedback that data has been submitted by displaying a simple message, or providing meaningful error messages if a mistake has been made.

KEY TERM

haptic: a form of computer interaction based on touch

Efficiency

Ensuring that users can accomplish their tasks quickly and with minimal effort helps to provide a positive user experience. Reducing the number of steps needed to complete a task, providing shortcuts for experienced users and ensuring that the system performs and responds quickly all contribute to system efficiency.

Learnability

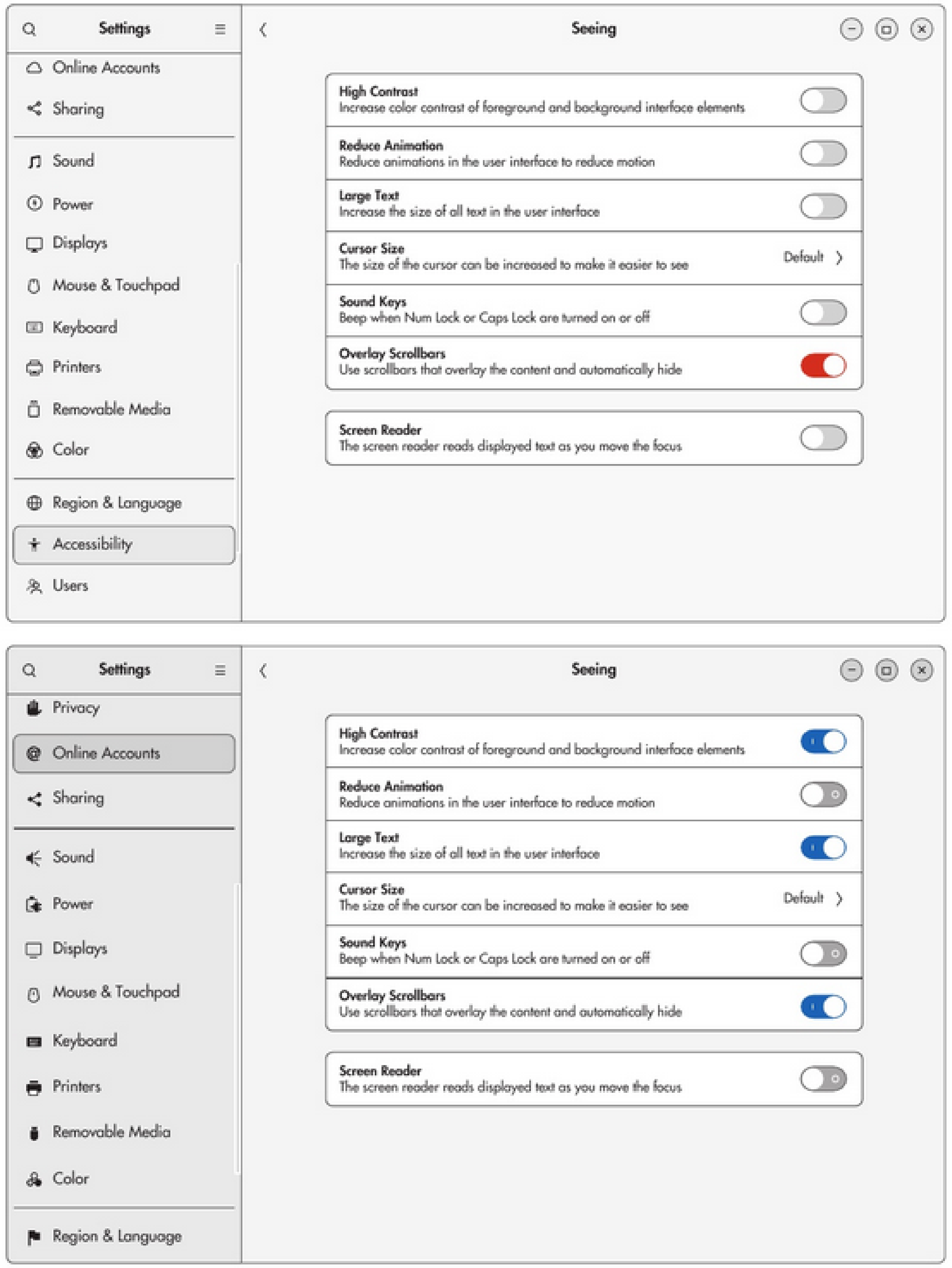
A system should be easy for users to learn. Clear and concise instructions, as well as intuitive design, contribute to the learnability of a system. Enabling a user to quickly get started with using a system, particularly core functionality, helps ensure that the user will keep using it and be more willing to spend time learning how to use the system for more complex functionality.

Accessibility

Ensuring that the interface is accessible to all users, including those with diverse needs, is an important factor in user experience. This involves designing interfaces that consider factors such as colour contrast, font size, keyboard navigation and compatibility with screen readers. The system should provide the user with the ability to apply and adjust these features as required, based on their needs (see Figure 13.1). Depending on how and where a system is to be used, there may be regional guidance and laws on how these features should be implemented.

Flexibility and customisation

Providing users with options to customise the interface according to their preferences, even for those users who do not have specific additional needs, enhances the overall user experience. However, it is essential to balance customisation with simplicity to avoid overwhelming users with too many choices.



High Contrast – Increase color contrast of foreground and background interface elements ○
Reduce Animation – Reduce animations in the user interface to reduce motion ○
Large Text – Increase the size of all text in the user interface ○
Cursor Size – The size of the cursor can be increased to make it easier to see – Default ›
Sound Keys – Beep when Num Lock or Caps Lock are turned on or off ○
Overlay Scrollbars – Use scrollbars that overlay the content and automatically hide ●
Screen Reader – The screen reader reads displayed text as you move the focus ○

High Contrast – Increase color contrast of foreground and background interface elements ●
Reduce Animation – Reduce animations in the user interface to reduce motion ○
Large Text – Increase the size of all text in the user interface ●
Cursor Size – The size of the cursor can be increased to make it easier to see – Default ›
Sound Keys – Beep when Num Lock or Caps Lock are turned on or off ○
Overlay Scrollbars – Use scrollbars that overlay the content and automatically hide ●
Screen Reader – The screen reader reads displayed text as you move the focus ○

▶ Figure 13.1 Accessibility options allow the user to make adjustments to the interface to suit their needs

SKILLS

CRITICAL THINKING, INTERPRETATION, COLLABORATION

ACTIVITY 4

It is important that a digital product is inclusive and accessible for all.

The World Wide Web Consortium (W3C) provides guidance on accessibility for web-based products and content. Explore their website to become familiar with the guidance; these principles are good practice for all computer-based systems and not just web content.

Go to the W3C website and complete some further research on accessibility guidelines. Make notes on your findings and discuss them with a partner.

KEY TERM

visual hierarchy: the order in which elements are noticed and their perceived importance

Aesthetics

Aesthetics refers to the appreciation of how pleasing something is to look at. While this can be subjective, a visually appealing design can positively affect the user experience. It involves considerations such as colour schemes, **visual hierarchy** and the overall look and feel of the interface.

SKILLS

INTELLECTUAL INTEREST AND CURIOSITY, ETHICS

ACTIVITY 5

Don Norman is a research professor focused on product design. He is best known for ‘Don Norman’s Principles of Interaction Design’, which consider how the design of a product affects user experience (UX).

These principles apply to many products, but they are often considered and used by those who design and create information technology systems.

Carry out some independent research into ‘Don Norman’s Principles of Interaction Design’ and how these principles can be used to inform the design of an information technology system.

Produce a short presentation about your findings.

IMPLEMENTING HCI

Although systems vary in their design and purpose, and the technologies that we use them on progress, the ways in which we interact with computers are quite consistent and fall into three broad categories:

- Visual
- Audio
- Haptic.

Visual HCI

Humans have a natural and powerful visual perception; visual HCI focuses on using this trait for users to interpret and interact with visual information in computer systems. At a basic level this may rely on simple text instructions, but effective visual HCI will make use of a wider variety of visual elements to communicate meaning and function in an intuitive way.

Design elements

There are a number of visual design elements that can be used to help ensure an effective interface, including the following:

- Layout: the way that elements such as buttons, menus and content are arranged on the screen can affect how the screen is interpreted and perceived. For example, some users may perceive something placed at the top of the screen as more important than something placed in the middle of a paragraph of text. If there does not appear to be a logical approach to how the content is laid out, this can create a sense in the user that the system is of poor quality, or it can make the information more difficult to understand.

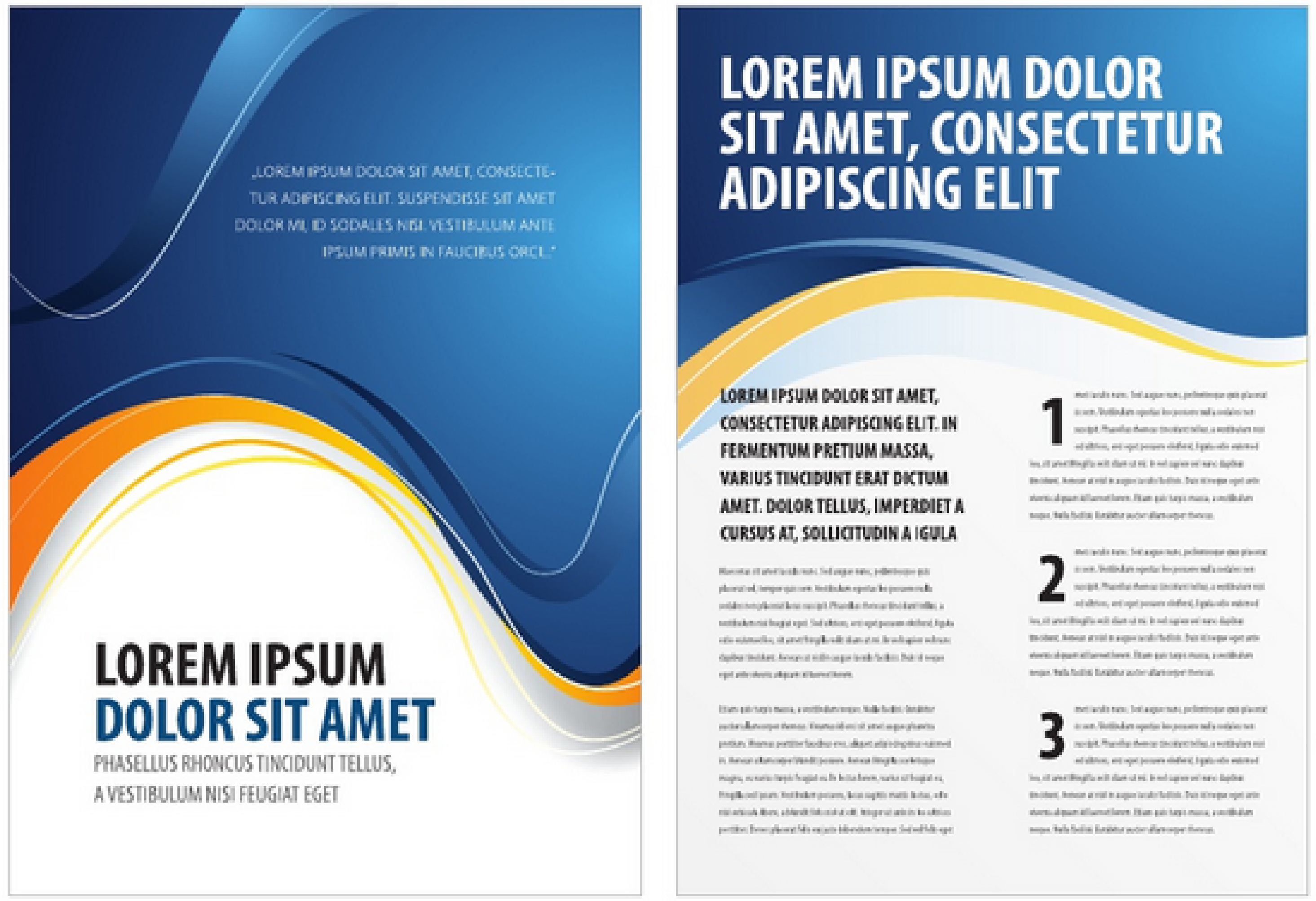
- Colour: effective use of colour can be used to convey information, emphasise elements and create a visually cohesive design. Colour can create a sense of identity; for example, you may associate the colours red and white with Coca-Cola®, or colours can evoke feelings such as the association of white and blue with the feeling of cold.

- Typography: the selection of fonts, font sizes and styles can enhance readability and convey a specific tone or style. Larger fonts, for example, can emphasise more important information.

- Images and icons: the inclusion of relevant images, icons and other visual cues can help recognition and provide a more engaging user experience. Including images is helpful in many ways; for example, a chart or graph can help the user interpret numerical data more easily, and a diagram or illustration can help support an explanation of a complex process.

Visual hierarchy

The sizing and placement of text is a key way in which visual information can be communicated. We are naturally drawn to larger text first, and we perceive this as being more important than smaller text. In a computer system there are many ways that this may be used; for example, buttons or links to core or important functions may be made more prominent than other, less frequently required features. Establishing a hierarchy of visual elements is important to guide users' attention and help them understand the relative importance of different components on the screen (see Figure 13.2).


„LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. SUSPENDISSE SIT AMET DOLOR MI, ID SODALES NISI. VESTIBULUM ANTE IPSUM PRIMIS IN FAUCIBUS ORCI."
LOREM IPSUM
DOLOR SIT AMET
PHASELLUS RHONCUS TINCIDUNT TELLUS, A VESTIBULUM NISI FEUGIAT EGET
LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT
LOREM IPSUM DOLOR SIT AMET, CONSECTETUR ADIPISCING ELIT. IN FERMENTUM PRETIUM MASSA, VARIUS TINCIDUNT ERAT DICTUM AMET. DOLOR TELLUS, IMPERDIET A CURSUS AT, SOLLICITUDIN A IGULA
1
2
3

▶ Figure 13.2 The visual hierarchy; which parts of the page are you drawn to first?

Responsive design

Modern systems, particularly web-based products, are likely to be accessed on a range of different devices such as smartphones, tablets and computers. Responsive design focuses on ensuring interfaces are designed in such a way that the size and layout of elements adapt to the different screen sizes and resolutions of the device they are being accessed on to ensure a consistent and usable experience.

Visual feedback

Providing users with visual cues can help ensure a better user experience. For example, providing a tick next to a data entry box to show that the data meets the format requirements, or displaying progress indicators for a particular task, helps the user interpret how a computer system is responding to their actions.

Audio HCI

Audio interaction can be used to enhance the user experience, provide additional functionality or support users with specific needs. Audio HCI focuses on the use of sound, speech and other auditory cues to provide information to users and allow alternative methods for the user to communicate with the system.

Speech interfaces

Traditionally, speech recognition systems required significant time investment from the user to train the system to recognise their voice, and the systems were often quite inaccurate, especially if the user had a regional accent.

However, in recent years the developments in machine learning have given rise to improved quality in natural language processing and speech recognition, allowing users to interact with a system through spoken language much more easily. These developments have given rise to widespread use of virtual assistants such as Siri, Alexa and Google Assistant, with many people using these as part of everyday life in their smartphones or devices in their home such as smart speakers.

Being able to control a system with your voice helps users who have limited technical skills, limited mobility or visual impairments to interact easily with the device without the need to read a screen or type in commands.

The use of speech synthesis allows the system to provide feedback to the user in auditory form as if holding a conversation. This can provide distraction-free feedback, such as providing directions when driving a car, or supporting those with visual impairments by reading a passage of on-screen text.

Auditory feedback

Using sounds or auditory cues to provide feedback on user actions, system status or the outcome of an operation, can enhance user experience. Auditory feedback is crucial for users who have visual impairments, or for situations where visual attention is focused elsewhere. Common auditory feedback includes:

- notifications, such as providing an auditory cue when a new email or text message is received
- alerts and warnings, such as collision detection warnings in a car
- sound effects, such as a click sound to notify the user that a virtual button has been pressed.

Haptic HCI

Haptic HCI focuses on the design and creation of interfaces that engage our sense of touch to create more natural, immersive and engaging experiences. The use of haptics can provide a number of benefits in a system and its users, including:

- increased realism and immersion by making digital experiences feel more natural and engaging
- improved learning and retention by engaging more than one sense so that the way we process information and remember tasks is enhanced
- improved accessibility by providing alternative or complementary interaction methods for people with visual or motor impairments
- greater emotional impact by creating a more engaging experience.

Force feedback

You may have experienced haptics through force feedback devices. These devices create physical sensations of pressure, vibration or movement using motors, actuators or other mechanisms, and they are commonly used in game controllers. This type of feedback is also often used in smartphones, where the phone's vibration function is used in conjunction with other actions to alert the user that something has occurred.

Tactile interaction

Tactile interaction provides an alternative communication method between users and a computer system. For example, the use of tactile displays allows visually impaired users to receive computer output in the form of Braille.

More recently, developments in technologies have integrated tactile output into augmented and virtual reality systems, providing more realistic interaction between the user and the system. Applications of this technology include medical simulations that allow doctors to train or practise medical procedures without using actual patients.

Biometric sensors

The field of haptics also includes other physical interaction such as movement. The rise in popularity of smartwatches and fitness trackers has led to an increase in the use of these to provide alternative interaction with other systems. For example, the heart rate data from a watch provides exercise and health data to an app, which can provide analysis of progress or alert the user to potential health problems.

THINKING LIKE AN IT SPECIALIST

In recent years there has been an increase in the number of health insurance companies that use data from smartwatches and fitness trackers worn by customers.

How might this data be used by insurance companies and what risks and challenges does its use pose?

SKILLS
ADAPTIVE LEARNING, ANALYSIS

ACTIVITY 6

A developer is creating an app to support children aged 6 to 10 years to improve their mathematics and reading skills.

Make a list of the ways that the developer could make use of different forms of human computer interaction to enhance the user experience and engage users.

Discuss your list with a partner and compare your ideas. Did your partner come up with things you hadn't thought of?

ERGONOMIC PRINCIPLES OF HCI

Ergonomics focuses on the design and arrangement of objects and environments to ensure that the interaction between users and computer systems is efficient, comfortable and productive. Effective application of ergonomic principles should take into account both the physical and cognitive limitations of humans and apply them to improving the interaction between the user and the computer.

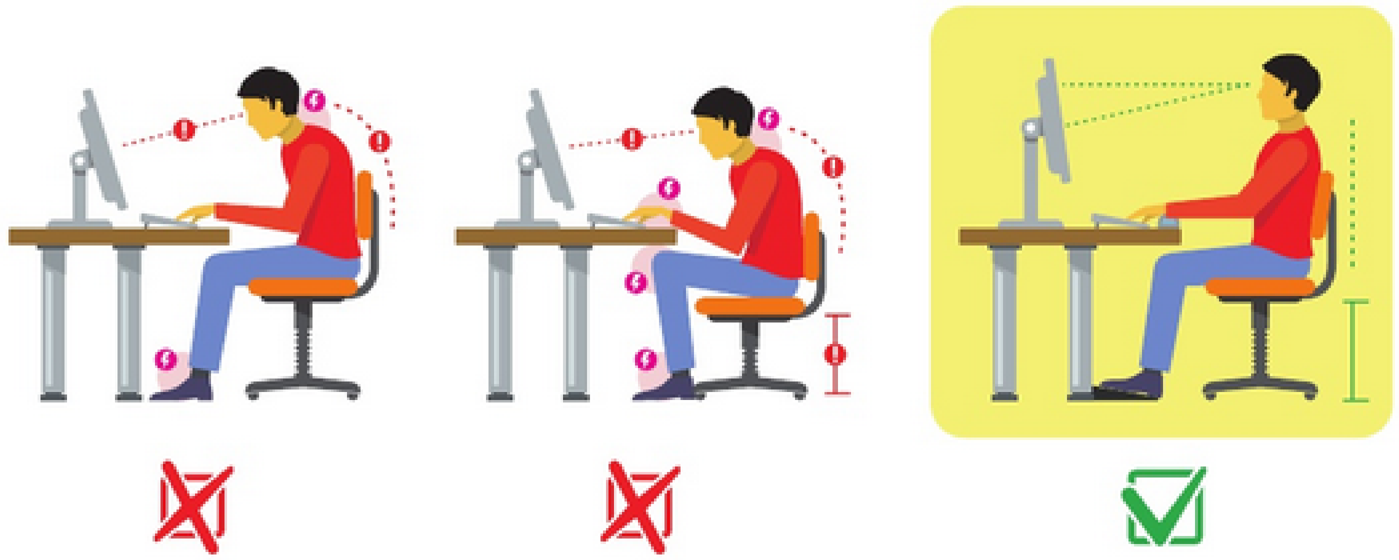
Physical ergonomics

Physical ergonomics is probably the most common area that people associate with ergonomics. This area focuses on physical aspects of human performance and well-being, including the user's body posture, movement and physical workload. The aim is to design workspaces, tools and equipment so that the physical stress, strain and fatigue on the user are minimised. This in turn reduces the risk of musculoskeletal disorders (MSDs).

Posture

Proper posture is crucial for avoiding MSDs like neck pain, repetitive strain injuries and back pain (see Figure 13.3). HCI principles provide a number of suggestions for users including:

- adjustable furniture to ensure that the user can sit correctly with a body position that does not apply undue stress to any single part of the body
- comfortable keyboard and mouse positions to avoid stress on the wrists, arms and shoulders.



▲ Figure 13.3 Adjusting the position of the computer and chair is important to avoid injury

Input devices

The design of keyboards, mice and other input devices should minimise stress on muscles and tendons. This includes features like ergonomic keyboards with curved layouts or the use of integrated wrist supports to ensure that the user's hands and arms remain level with the keyboard. The use of trackballs instead of mice can also reduce the strain on the user. As the accuracy of voice recognition software has improved, this can also be used to reduce the amount of physical interaction a user has to make with a computer.

Visual display

Prolonged or improper use of a computer display screen can cause eye strain and headaches. Display screens should have adjustable brightness, contrast and resolution so that eye strain can be reduced. Provision should be made to ensure text size and font styles are legible, either through use of an appropriately sized screen, or through features that allow the user to alter the size and zoom level to suit their needs.

Another key consideration is the location of a system in relation to any light sources. Glare on a screen reduces the user's ability to see correctly and increases eye strain. The amount of glare should be reduced, for example, by:

- placing the computer at right angles to windows
- using shades on windows to reduce the brightness from outside light sources
- adjusting the lighting in the room.

SKILLS

INVESTIGATION, ANALYSIS

ACTIVITY 7

Many countries have laws or regulations that provide guidelines for organisations about the safe and proper use of computer systems.

Investigate the law and guidelines in place in your country.

Produce a set of notes on what you have found out. Discuss your notes with a partner, then spend a few minutes updating your notes based on your discussion.

Cognitive ergonomics

In addition to physical demands, ergonomics covers the demands of interaction with a computer system on a conscious intellectual level. These principles influence the way the system is designed to ensure that the interface and processes are efficient and effective.

Workload

Interfaces should be designed to minimise cognitive workload, which can lead to fatigue, errors and reduced productivity. This can be achieved through the way in which information is presented. Reducing the amount of information that is given to the user at any one time, or effective use of graphics, can assist with this. Other ways include keeping interfaces simple, consistent and predictable.

Memory

Human memory is limited so interfaces should use familiar patterns and provide reminders and cues to help users remember how to perform tasks.

Attention

Interfaces should avoid distractions and help users focus on the task at hand. Minimising unnecessary visual clutter and auditory cues can assist in reducing distractions. Reducing alerts, pop-ups and adverts can help this, as can adding features such as 'do-not-disturb', which allows users to limit the types of alerts that they receive when trying to focus on a task.

THINKING LIKE AN IT SPECIALIST

The International Organization for Standardization (ISO) has a series of standards related to ergonomics of human computer interaction (ISO 9241).

Investigate ISO further; what other standards can you find that contribute to the safety and well-being of the users of computer systems?

DESIGNING AN INTERFACE TO MEET THE NEEDS OF INDIVIDUALS AND ORGANISATIONS

Although graphical user interfaces (GUIs) may vary in an aesthetic manner from system to system, there are some common features and principles that should be applied to ensure that an interface meets the needs of the individual user and the organisation for which the system is designed.

In the Implementing HCI section, you learned about specific design principles for improving the quality and effectiveness of an HCI design. In this section, you will build on that knowledge by exploring how those principles can be applied to common features of GUI design.

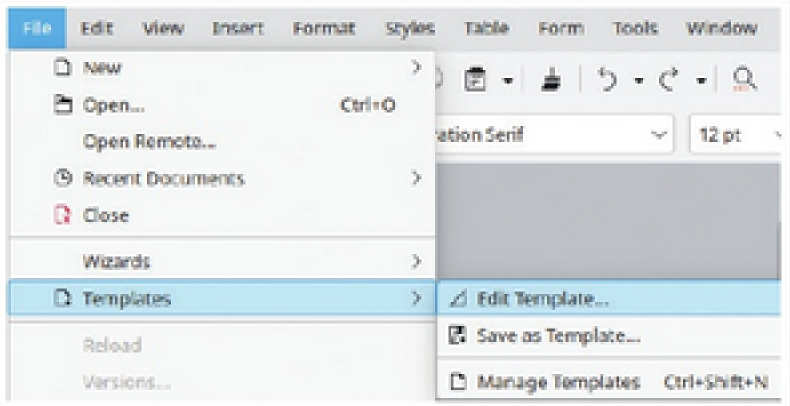
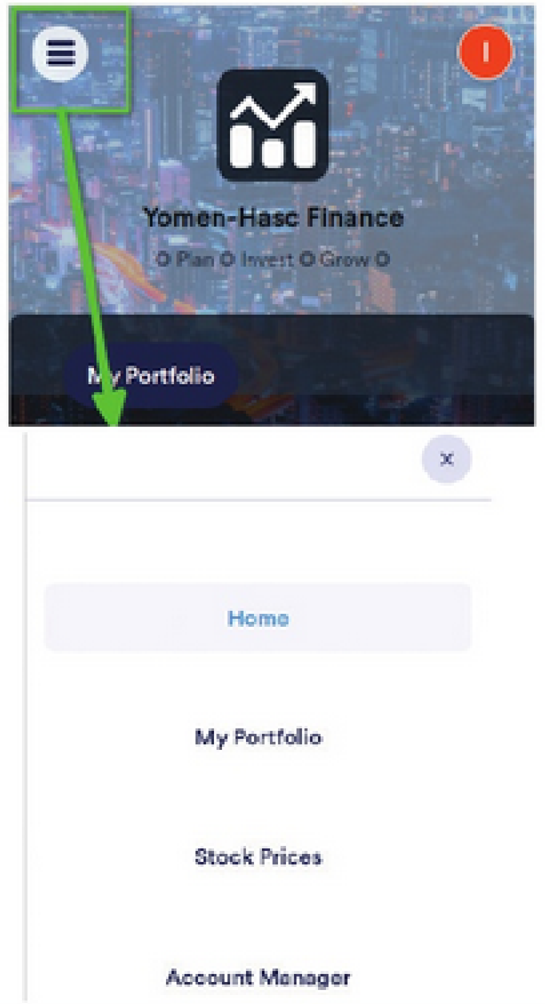
Menus

GUI menus are one of the most common and well-known features in modern software. Menus allow quick access to tools and features without the need for the user to remember strings of typed commands.

A well-designed menu system will be intuitive and will group related features together. When developing a new system, while it is important to be creative to make the system appear new, exciting and stand out from the competition, it is also important that the user feels it is familiar enough that they are not dissuaded from using it. In menu systems, some conventions have been established, for example, actions such as 'Save', 'Open' and 'Print' will almost universally be found in the 'File' menu.

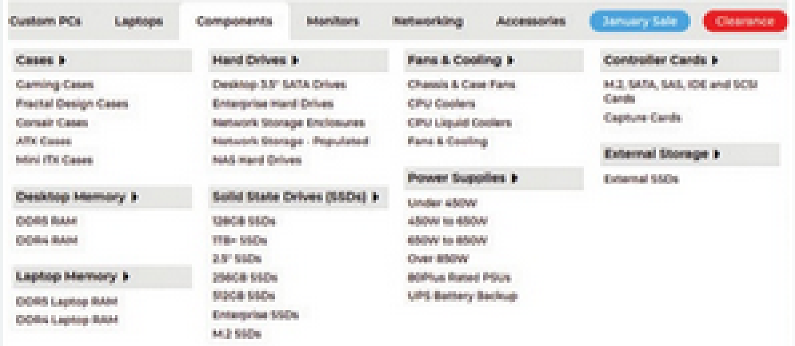
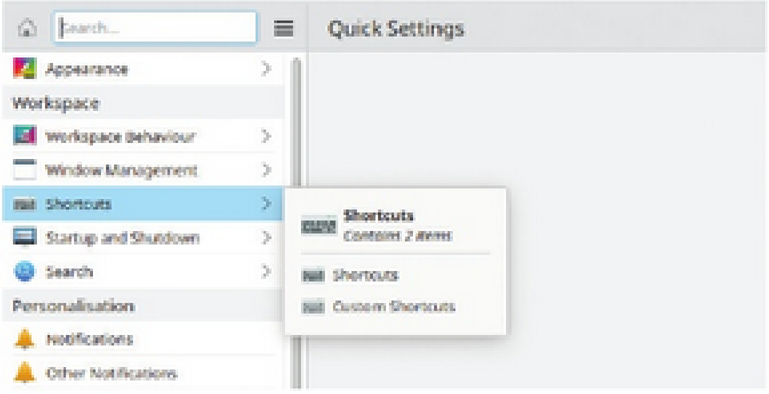
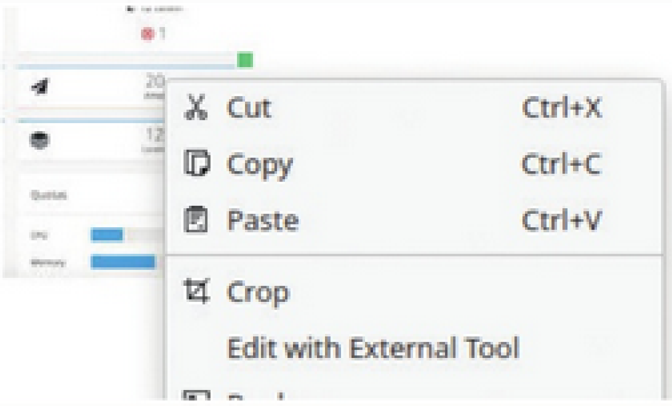
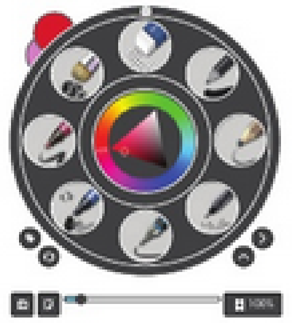
However, there are a number of ways that you can make a menu system creative and appealing. Table 13.2 shows some examples of different types of menu that are common, and provide the designer with a range of options to make a system both usable and appealing.

TABLE 13.2: MENU TYPES: WHICH OF THESE HAVE YOU USED?

Menu type	Description	Example
Standard horizontal menu	Typically found at the top of a software or application window. Each menu contains a series of related options, which can lead to submenus.	 File Edit View Insert Format Styles Table Form Tools Window New Open... Ctrl+O Open Remote... Recent Documents Close Wizards Templates Reload Versions... Edit Template... Save as Template... Manage Templates Ctrl+Shift+N ation Serif 12 pt
Hamburger menu	Usually located in a corner of a screen, a hamburger menu expands when selected but hides the options when not in use. It is very useful on smartphones or tablets where smaller screen sizes mean less can be shown on the screen.	 Yomen-Hasc Finance O Plan O Invest O Grow O My Portfolio × Home My Portfolio Stock Prices Account Manager

(Continued)

TABLE 13.2: MENU TYPES: WHICH OF THESE HAVE YOU USED? (*Continued*)

Menu type	Description	Example
Mega menu	A large multi-column menu that is hidden until one of the top-level items is selected or hovered over with the mouse. Typically used where there are many possible options or subcategories, such as on an e-commerce site.	 Custom PCs Laptops Components Monitors Networking Accessories January Sale Clearance Cases Gaming Cases Fractal Design Cases Corsair Cases ATX Cases Mini ITX Cases Desktop Memory DDR5 RAM DDR4 RAM Laptop Memory DDR5 Laptop RAM DDR4 Laptop RAM Hard Drives Desktop 3.5" SATA Drives Enterprise Hard Drives Network Storage Enclosures Network Storage - Populated NAS Hard Drives Solid State Drives (SSDs) 128GB SSDs 1TB+ SSDs 2.5" SSDs 256GB SSDs 512GB SSDs Enterprise SSDs M.2 SSDs Fans & Cooling Chassis & Case Fans CPU Coolers CPU Liquid Coolers Fans & Cooling Power Supplies Under 450W 450W to 650W 650W to 850W Over 850W 80Plus Rated PSUs UPS Battery Backup Controller Cards M.2, SATA, SAS, IDE and SCSI Cards Capture Cards External Storage External SSDs
Vertical sidebar navigation	A menu bar positioned on the left or right side of the screen and usually arranged in a vertical manner. Will often use hidden submenus that are only shown when a main option is selected. Useful for complex applications with a lot of features.	 Search... Quick Settings Appearance Workspace Workspace Behaviour Window Management Shortcuts Startup and Shutdown Search Personalisation Notifications Other Notifications Shortcuts Contains 2 items Shortcuts Custom Shortcuts
Contextual menu	A menu that appears dynamically when the user right-clicks on a specific element within an application or web page.	 Cut Ctrl+X Copy Ctrl+C Paste Ctrl+V Crop Edit with External Tool
Pie menu	A circular menu with options that expand outwards. Common in graphics editing and design applications. These can be customised by the user to provide access to frequently used tools.	

SKILLS

PROBLEM SOLVING, CREATIVITY, COLLABORATION

ACTIVITY 8

Find examples of applications and/or websites that use the menu types listed in Table 13.2.

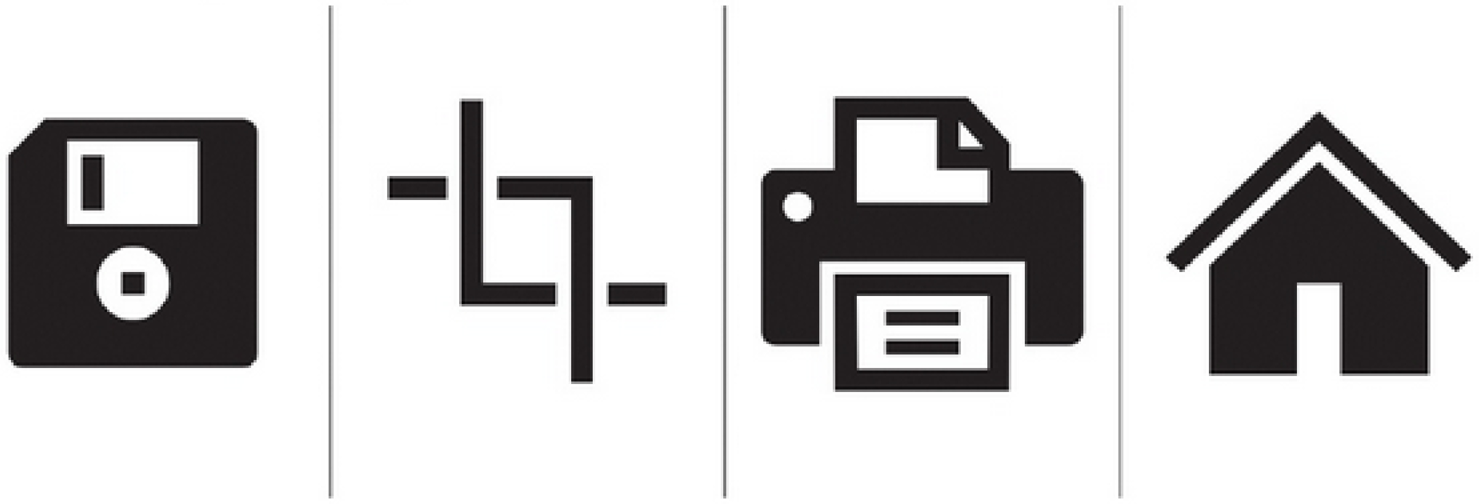
With a partner, produce a slideshow about how they help the user access the content and navigate to information or tools.

When you have completed your slideshow, join with another group and present your slideshows to each other.

Icons

Icons are small pictures that act as shortcuts to files, programs or software features in a computer (see Figure 13.4). Icons improve accessibility and ease of use by using clear and simple images to represent a concept without the need to read text.

Icons can help provide consistency within a suite of applications by using the same style, size and colours. They also increase familiarity when using a new application because certain icons are commonly used to represent the same actions.



▲ Figure 13.4 Can you name the function of each of these icons?

Accessibility

As you covered in the section on implementing HCI, accessibility is an important factor that must be considered to ensure inclusion of all potential users. There are a number of ways to incorporate accessibility into a user interface. These include the following:

Adjustable typeface: allowing the user to change the style and size of the font enables users who have additional visual needs to access the content more easily.

Adjustable colour schemes/high-contrast option: some items can be more difficult to see for some users if they are in certain colour combinations. Providing the option to change the colours of fonts and backgrounds, or providing an option that sets the page to a high-contrast option, allows users to change the page into a format that makes each item more distinct from surrounding items.

Screen reader support: a screen reader is a piece of software used by those with impaired vision to access on-screen content. The screen reader, as the name suggests, picks up any text embedded on a page or screen and uses speech synthesis to read it to the user. As well as the main text, the screen reader will identify types of content such as buttons, tags and labels, and read these to the user. When designing a page for screen reader support, you should organise the information in such a way that it is logical and, where possible, add meaningful metadata so that the screen reader can identify the purpose of an object.

Alt text: alt text is metadata that should be provided with any buttons or images used on a page. Alt text should provide meaningful information that can be read by the screen reader and help the user. For example, the alt text for an image should provide a meaningful description of the contents of the image.

Captions and transcripts: if a video or other multimedia content is included, it is good practice to provide captions/subtitles on the video to support users with impaired hearing, or those who are accessing the content while in a noisy environment. Transcripts provide a text version of spoken words and other audio elements, providing an alternative way for users to access the content.

Windows

Windows are a graphical component of a computer screen interface and are common to most programs and applications. Nearly all modern operating systems such as Windows, macOS and Linux, make use of windows to group and manage content and data. Most operating systems allow the windows to be customised to a greater or lesser degree, and most have these key characteristics:

Rectangular frames: each window has a rectangular border that visually separates it from other elements on the screen.

Title bar: the bar at the top of the window typically displays the name of the program or document being used. This allows the data to be identified by the user or by screen readers, or by other accessibility features.

Buttons: buttons will often be present for common actions like closing, minimising and maximising the window.

Menu bar: some windows have a menu bar at the top that provides access to additional features and functions.

Scroll bar: if the amount of content does not fully display at the window's current size, a scroll bar will often appear. Typically, a scroll bar will appear at the right-hand side for vertical scrolling and at the bottom for horizontal scrolling.

Content area: this is the main space within the window where the actual program interface or document is displayed.

Pointers

Pointers, also referred to as cursors or mouse pointers, show where the user's current interaction is on the screen. The pointer represents the interaction between the user and the screen when using devices such as a mouse, trackpad, stylus or even their finger.

It is common to represent the pointer as a form of arrow. The pointer is often contextual in that its appearance will change depending on the type of action the user is performing.

SKILLS

CREATIVITY, INTERPRETATION, PROBLEM SOLVING

ACTIVITY 9

Yomen-Hasc Finance is a small savings and investment company that provides help and advice to customers about their finances.

The company is updating its website. The website will contain an interactive feature that will provide customers with advice about investing in stocks and shares.

It will contain a number of different screens including:

- general investment advice
- a search tool that allows users to find information about a specific company or stock they may wish to invest in
- a tool that will allow the user to plan a budget for savings and investment. The tool will provide feedback on how much money they would have after a given amount of time.

Create a design/plan for a suitable interface for this section of the website and the interactive feature.

You may wish to create the design on a computer, or you can draw it by hand.

Your design should demonstrate an understanding of HCI principles and contain sufficient detail that a website developer could use it to create the final product.

Use the principal outlines in the W3C guidelines and the tools on WebAIM.org to help you.

THINKING LIKE AN IT SPECIALIST

WebAIM.org is an online service that allows web designers to check the accessibility of their chosen design. The service allows users to select colours and font combinations, which are then evaluated and given an accessibility score.

When you design an interface, use this or a similar service to check that your design is accessible.

STORING AND SECURING DATA

All computer systems are built on data. Whether it is the program code that defines how a computer program runs, the basic instructions that control how the hardware operates or the information the system will access and use, these all exist as data.

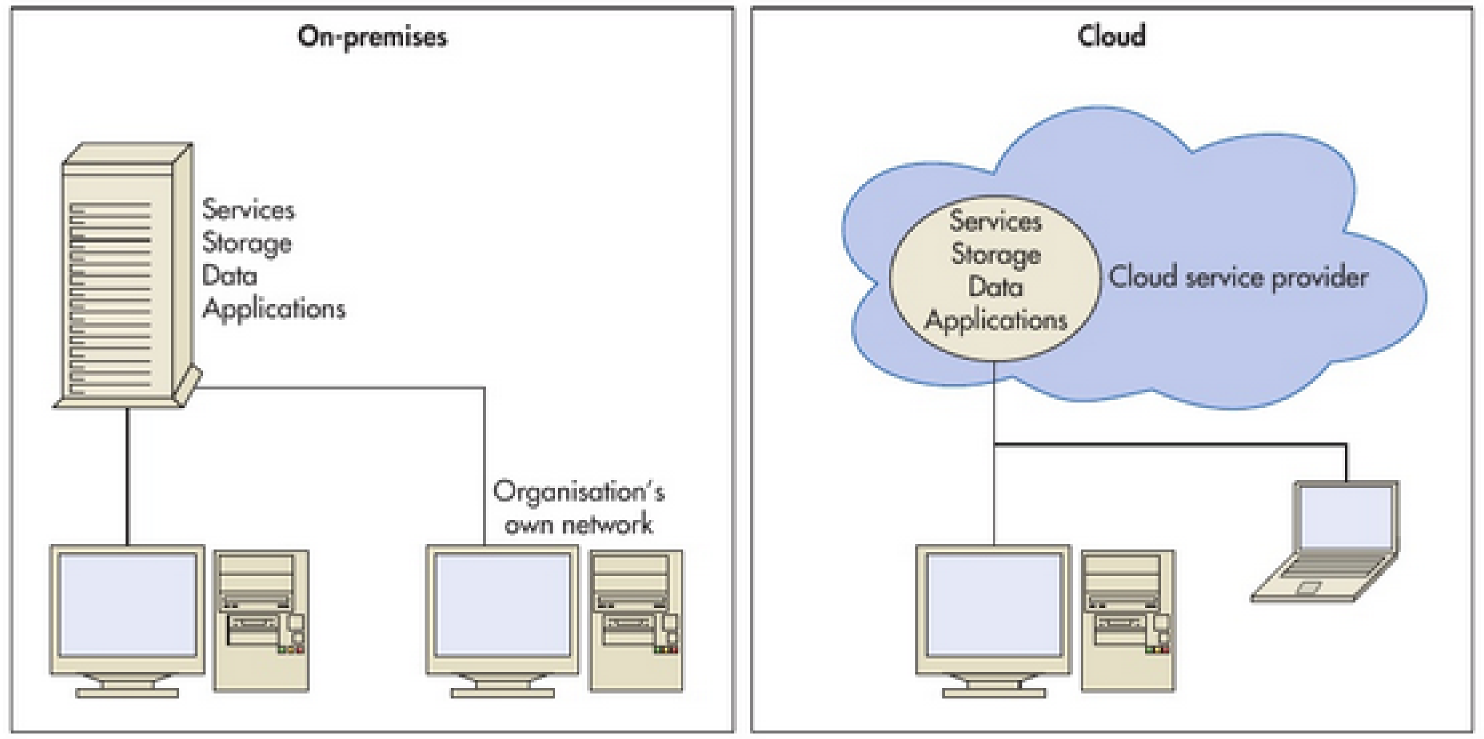
In order to make effective use of data, an IT professional needs to carefully consider how and where the data will be stored so that it is easy to access for those that need it, but restricted for those who shouldn't have access.

STORING DATA IN THE CLOUD

The term 'cloud' is used to represent any tool or resource that is provided by a third party and accessed via the internet.

Traditionally, companies would store all their data on computers and servers that were kept and maintained in their own buildings. This is referred to as 'on-premises' resources. While many companies still do this, many have moved to using the cloud for some or even all of their data storage needs.

When storing data 'in the cloud', the data is still held on a server. However, this server is not in the same location as the user (see Figure 13.5). The hardware that the data is stored on is maintained by a cloud services provider such as Amazon Web Services or Microsoft Azure, and companies or individuals pay a fee for access to these services.



▶ Figure 13.5 A comparison between the cloud and 'on-premises'

SECURING DATA

Data is extremely valuable: it can give key insights to inform business decisions that can lead to increased profits, or advantages over rivals. It can also be sensitive, so ensuring that data is secure to avoid its loss or misuse is vital for companies and individuals.

There are many ways to protect data, and the method that you use can depend on many factors. However, one of the key factors is whether the data is stored or being transmitted. In this section we will explore some of the most common methods of securing stored data.

File encryption

The term 'file encryption' is typically used to refer to encryption of stored data, rather than data during transmission, which typically omits the use of the word 'file'. However, these terms can be used interchangeably, as the basic principle is the same. You will explore how encryption is implemented for transmitted and stored data later in this topic.

At its simplest level, encryption is a security method that protects the data in a file from an unauthorised user by using an **algorithm** to scramble the contents so that it is unreadable without the correct key.

Although the algorithms used for encryption are themselves very complex, the overall process of encryption, in terms of the mathematical principles involved, is relatively simple. Versions of this process, also known as cryptography, have existed for thousands of years. At its simplest level, encryption uses a mathematical formula to change the contents of the file into an unreadable form (ciphertext). This process is controlled by a 'key', which is either a user-generated password or a randomly generated string. The data cannot then be unscrambled or read without the original key.

KEY TERMS

algorithm: a finite set of predefined steps to solve a problem

brute-force attack: persistent and repeated attempts to gain access to a system by guessing a password

SKILLS

INTELLECTUAL INTEREST AND CURIOSITY, REASONING/ ARGUMENTATION

ACTIVITY 10

Cryptography is an area of study in IT security.

Select a simple encryption method, such as a substitution cipher, and use it to send an encrypted message to a partner. What do think is the primary weakness in the exchange with your partner?

Use the internet to find out how the 'Enigma' machine worked. What were the strengths of the approach? What were the weaknesses? How was the code eventually broken?

Passwords

Passwords are a common form of security used to restrict access to computer systems. Passwords are used to authenticate the identity of a user through the supply of data or information that should only be known to that user. A computer system securely stores the password, which will have been provided by the user when the system was configured. When access is requested to a system or file, the user is challenged to enter the password. The entered data is compared to the stored data and, if they match, then access is granted.

When selecting a password, you should endeavour to make it as difficult to bypass as possible. Some ways that you can ensure your passwords are more secure are:

Complexity: a strong password is one that is difficult for others to guess or crack. This can be achieved by using a combination of uppercase and lowercase letters, numbers and special characters. You should also avoid using personal information such as a favourite sports team or pet name as these are easier for others to guess.

Length: longer passwords are generally more secure than shorter ones. It increases the number of possible combinations, making it more challenging for attackers to guess. A long password also means that it will take much longer to crack using **brute-force attack** methods. Many security guidelines recommend using passwords with a minimum length to enhance security.

Uniqueness: a unique password should be selected for each system and service used. This means that if an attacker gains access to one password, they will not be able to use it to compromise other accounts.

My Password 123456

▶ Are all the passwords you use for your digital accounts secure?

Passwords are a form of 'knowledge-based' authentication; i.e. you prove that you are the user by providing information that only you should know. There are a number of ways that knowledge-based authentication is implemented in computer systems including the following:

Passwords: often paired with a username; these are used to gain access to a computer system such as a computer OS or online service.

Security question: these are typically used to provide an extra layer of security to a user's account. Security questions use personal information to challenge the identity of the user. These are often used if a user has to prove who they are after forgetting their password, or if an unusual task is being carried out (e.g. a much more expensive purchase than usual).

PIN: a personal identification number is used to protect payment cards or access to smartphones.

Pattern lock: this is used to restrict access to a smart device; a pattern lock requires the user to draw a unique pattern on the touchscreen with their finger.

SECURITY IN DATABASE MANAGEMENT SYSTEMS (DBMS)

A database management system (DBMS) controls access and views of stored data so that data security, integrity and privacy are maintained.

Access control tools in a DBMS allow the data manager to control who can access the database, what actions they can perform once they have access, and what subset of the data they can view or modify.

Views are used to provide a way of presenting specific subsets of the data to users, hiding the complexity of the underlying database schema and limiting the number of potential points where errors or corruption can occur.

Access and views are controlled by the DBMS in a number of ways.

Authentication

A DBMS verifies the identity of users attempting to access the database in the same way as other computer systems authenticate users before granting access, so users must provide valid credentials, such as usernames and passwords.

Authorisation

Authorisation involves granting or denying permissions to perform specific operations (e.g. read, write, delete) on certain database objects. In an organisation, there may be a large number of people who need access to a particular set of data, but their needs may vary. For example, some jobs may just require a user to find and read data to locate names and addresses whereas another role may require the ability to edit certain data. User roles and privileges are closely linked with authorisation, and allow authorisation to be fine-tuned so that only the data a user should access can be accessed.

Roles: a DBMS allows the database manager to define roles that group together certain sets of privileges. For example, the sales team would have a different set of privileges from senior managers. The security settings for each role can be defined once but applied to all users in that role, which reduces the possibility for a single user to be assigned privileges that they should not have.

Privileges: these will specify the types of operations a user or role is allowed to perform, such as limiting to read-only, or allowing editing of certain data points. A guiding principle in data security is that of 'least privilege', that is to say, a user should be given just enough access to perform their job effectively and no more.

Other methods of control

Access control lists (ACLs) are lists associated with specific database objects (such as a table). These lists define the users or roles that have access to that object and the type of access they will have.

Row-level security (RLS) allows the database manager to control access to individual rows in a table based on certain conditions. For example, a user might be restricted to viewing only those rows where the name in the 'department' field matches their department.

Views provide a representation of the data in the database so that users see a specific subset of the data without exposing the underlying complexity of the database model itself. Views can be used to restrict access to specific data and present only the necessary information to users.

Auditing features track and log user activities, so that a record is kept of who accessed the database, what they did when accessing the data and when this action occurred. This provides a trail that can be used as part of any investigation following an issue such as a security breach.

ENCRYPTION

Encryption is a data security method that protects data from unauthorised users by rendering it unreadable to anyone but the permitted users. Using a mathematical algorithm, data is scrambled, and it can only be unscrambled by users who have the appropriate key.

Encryption falls into two main categories: symmetric encryption and asymmetric (also known as public-key) encryption. Typically, symmetric encryption is used to secure stored data, which will be accessed by very few people (for example, encrypting the hard drive on an individual's computer), whereas asymmetric encryption is more commonly used for protecting data during transmission, or data on remote systems (for example, accessing an online account or making an e-commerce purchase).

SYMMETRIC ENCRYPTION

Symmetric encryption uses the same key to both encrypt and decrypt the data. Whoever knows the key can access the data. Symmetric encryption is very useful for securing stored data that will have very few users requiring access, or for securing personal devices, such as a USB memory stick (see Figure 13.6). However, it is less effective for securing data that has to be transferred between computer systems, or shared with others, because of the need to share the key with whoever will be accessing the data.

The process of implementing symmetric encryption works in three stages.

Step 1 – Key generation: the key should ideally be a string of random letters and numbers, or sometimes a complex password. It is good practice to use a computer program to generate a complex and unique key so that it is much harder to guess or brute-force attack.

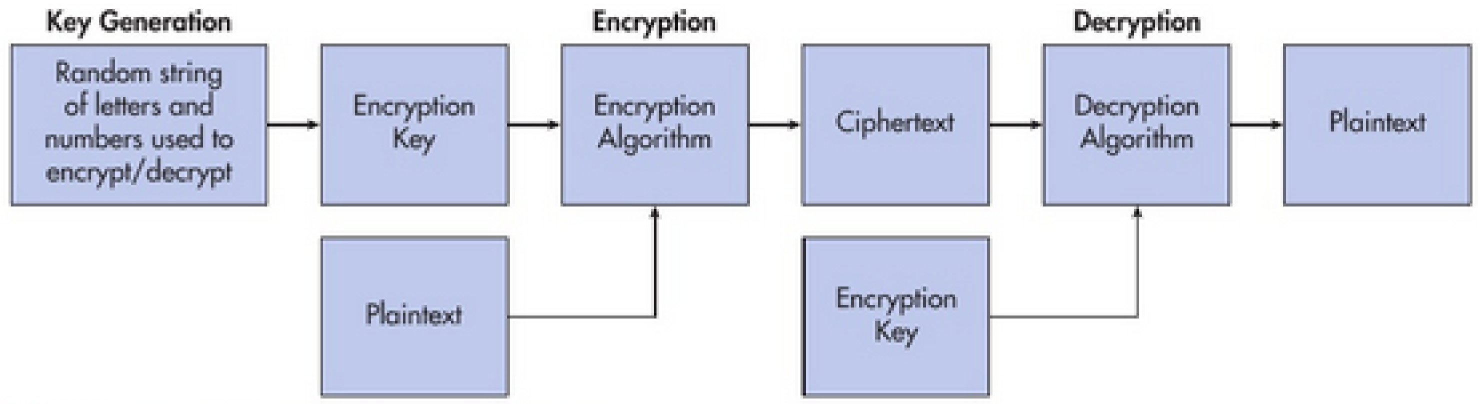
KEY TERMS

ciphertext: the encrypted version of a message/data

plain text: the original and/or decrypted version of a message/data

Step 2 – Encryption: the original data (**plain text**) along with the key are input into an encryption algorithm. This algorithm uses the key and mathematical processes to scramble the plain text so that it becomes unreadable. The resulting encrypted data (**ciphertext**) is unintelligible without the key.

Step 3 – Decryption: the ciphertext and the same key that was used to encrypt the data are applied to a decryption algorithm. The decryption algorithm reverses the encryption process to convert the ciphertext back into plain text.



▲ Figure 13.6 The symmetric encryption process

Benefits and limitations of symmetric encryption

Symmetric encryption provides excellent security from opportunistic attacks. For example, if your laptop is stolen, having the data encrypted will provide excellent protection for any data stored on the device. The security of symmetric encryption relies heavily on the security of the key, and because the criminal is highly unlikely to have also obtained the key, the data will be very safe. Symmetric encryption is usually a much faster process than asymmetric encryption, and uses a less complex algorithm. Therefore, when encrypting large amounts of stored data or using less powerful personal devices, symmetric encryption is a more efficient option.

The main challenge with symmetric encryption, especially if using it for shared or transmitted data, is ensuring that the key remains secure. First, you need to share the key with all the users who need the data. The key is vulnerable when it is being exchanged, so it is often shared using key exchange protocols, which use symmetric encryption to securely exchange symmetric keys.

The second challenge is keeping the key secure once it has been shared. Because the same key is used by everyone who will access the data, the more people who have access to the key the greater the likelihood that it will be accidentally, or intentionally, shared with someone who should not have access.

ASYMMETRIC ENCRYPTION

Asymmetric encryption, also known as public-key encryption, is a highly secure form of encryption that uses two related keys for the encryption and decryption process (see Figure 13.7). The two keys, referred to as the 'public key' and the 'private key', are mathematically related but different from each other and each serves a separate purpose.

The public key is used for the encryption part of the process. It can be widely shared and anyone can use it to encrypt messages intended for the owner of the related private key.

The private key is never shared and is used for decryption. This ensures that only the owner of the private key can decrypt messages that have been encrypted using the corresponding public key.

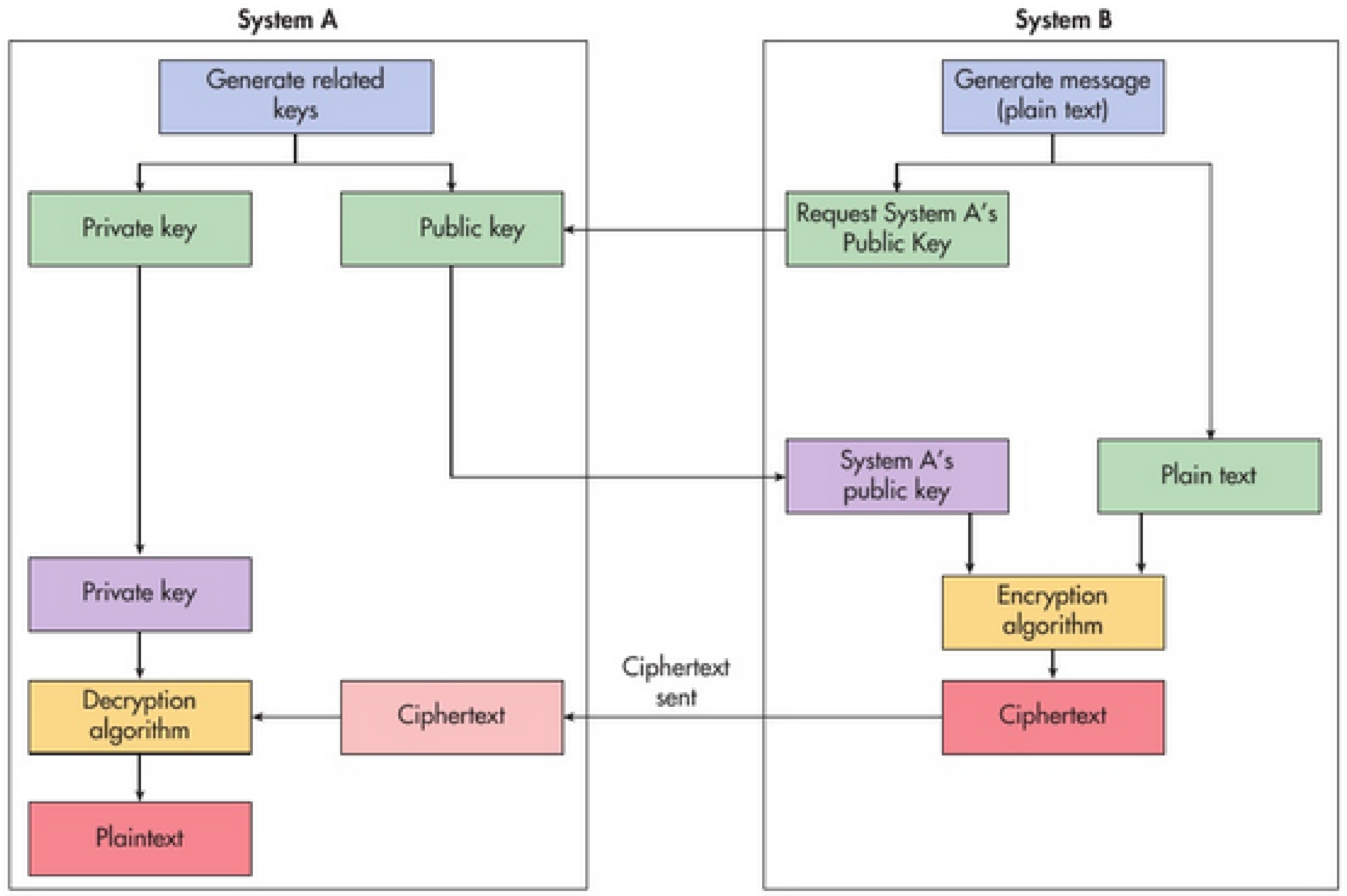
The process of implementing asymmetric encryption works like this:

Encryption stage:

- Step 1 – the sender obtains the recipient's public key.
- Step 2 – using the public key, the sender encrypts their plain text message to create the ciphertext.
- Step 3 – the ciphertext is then sent to the intended recipient.

Decryption stage:

- Step 1 – the recipient applies the decryption algorithm to the ciphertext using their private key.
- Step 2 – because the private key is mathematically related to the public key that was used for encryption, the decryption algorithm is able to convert the ciphertext back into plain text.



▶ Figure 13.7 The asymmetric encryption process

Benefits and limitations of asymmetric encryption

Public-key encryption is highly secure and allows data to be shared securely using potentially insecure channels (for example, sending personal data over the internet). Although they are mathematically related, the complexity of the algorithm used to generate the public and private keys means that it is mathematically improbable that the private key can be engineered from knowing the public key alone, which allows it to be easily and freely shared.

Because of the complex nature of the process, it is not typically used for securing large quantities of data, or for personal devices, because of the time and processing power needed.

CERTIFICATE-BASED ENCRYPTION

Certificate-based encryption is a system that uses digital certificates to authenticate the identity of those involved in the communication process, so that a trusted and secure connection can be established. It is used to add an additional layer of security when using asymmetric encryption.

The digital certificate is a digital document that is used to prove the identity of the owner of a public key. The certificates are issued by a trusted third party, known as a Certificate Authority (CA), who will check the validity and trustworthiness of an entity (such as an organisation, or specific server), and then issue the certificate as proof of identity.

The certificate contains specific information, which is exchanged between the sever and the client when establishing a secure connection.

Public key: the certificate provides the public key of the identified entity. As you learned in the previous section, the public key is used to encrypt information that can only be decrypted by the corresponding private key.

The subject: this confirms the identity of the entity to which the public key belongs. Similar to a passport or driving licence for a person, it confirms the identity of the person, organisation or website that is being communicated with.

Issuer: this is the name of the trusted third party (CA) that issued the certificate.

Validity period: this is the period of time during which the certificate is valid. This helps improve security as entities must regularly renew their digital certificates, which will involve the CA re-checking their validity and trustworthiness.

Digital signature of the issuer: this is a digital signature that verifies that the certificate was issued by the trusted third party.

KEY TERMS

HTTPS protocol: the secure version of the protocol used to transfer and render webpages

secure socket layer (SSL): a networking protocol that provides secure connections between web clients and web servers

Certificate-based encryption is an extension of asymmetric (public-key) encryption, and part of the process is largely the same. However, there are additional levels to add greater security, while still maintaining system efficiency (see Figure 13.8). Certificate-based encryption is commonly used by websites to protect customer data. This is used by the **secure socket layer (SSL)** in a website that uses the **HTTPS protocol**.

Take, for example, a typical exchange between a customer and an e-commerce site.

Stage 1 – certificate exchange:

The customer accesses the e-commerce site using their web browser.

When a secure communication channel is needed, such as when they need to enter payment information, the customer's browser will request that the server (e-commerce site) identifies itself.

The server will send a copy of its SSL certificate to the customer's computer.

The web browser checks the authenticity of the certificate by checking the CA's signature and confirming that the certificate has not been revoked.

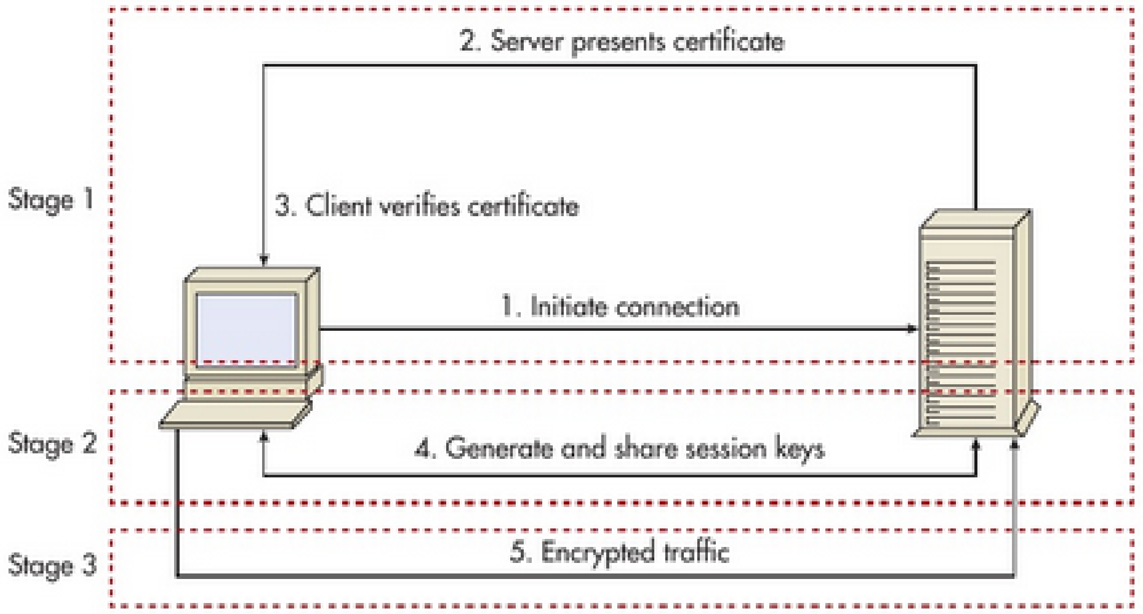
Stage 2 – key exchange and encryption:

When the validity of the certificate, and by extension the other entity's identity, has been confirmed, the browser creates a unique one-time session key (symmetric encryption), which is then encrypted using the server's public key.

The server then decrypts the session key using its own private key and sends back an acknowledgement that has been encrypted using the session key.

Stage 3 – data transmission:

Once the encrypted session is established, the server and browser can securely transmit data over the communication channel using symmetric encryption.


Stage 1
2. Server presents certificate
3. Client verifies certificate
1. Initiate connection
Stage 2
4. Generate and share session keys
Stage 3
5. Encrypted traffic

▶ Figure 13.8 Certificate-based encryption

Benefits and limitations of certificate-based encryption

Certificate-based encryption allows systems to securely exchange a unique session key. Because the session key encrypts data using symmetric encryption algorithms, data transfer is generally faster and more efficient than if asymmetric encryption was used for the whole process. This ensures that data is exchanged between the two entities without a loss in performance. This is particularly important for sites that experience a lot of traffic or want to ensure a good user experience.

There are a few minor limitations with certificate-based encryption, including the need to trust that the CA adequately checks the organisations that they issue certificates to, and that the CA itself is trustworthy.

However, the increased security provided by certificate-based encryption greatly outweighs the potential limitations, and it is widely used to provide secure communication in a range of systems, including:

HTTPS websites for secure online transactions and communication

email encryption to protect sensitive messages and information exchanged via email

software signing, which verifies software authenticity of software to reduce the chances of potential malicious code

VPN connections, which create secure tunnels to allow secure remote network access and protect data when communicating over the internet.

CHECKPOINT

1. What is virtualisation and how can it be achieved?
2. What is meant by a 'distributed system' and why are the systems needed?
3. What is human computer interaction?
4. What are the ways that human computer interaction can be implemented?
5. Why are ergonomic principles important when considering human computer interaction?
6. Why is it important to secure data on a cloud-based system, and how can that be achieved?
7. What role does a database management system (DBMS) play in controlling access to data?
8. How does encryption protect data, and how can encryption be implemented?

SKILLS

CRITICAL THINKING, REASONING

EXAM PRACTICE

1. Explain two ways that using virtual servers can save money for an organisation. [4 marks]
2. Explain one reason why containerisation provides better portability than virtualisation. [2 marks]
3. Concurrency is one potential issue with a distributed system.

   State two other potential issues. [2 marks]
4. Explain what is meant by concurrency in relation to a distributed system. [3 marks]
5. Describe how asymmetric (public-key) encryption protects data. [4 marks]

TOPIC 14
USING IT SYSTEMS IN ORGANISATIONS

09 : 09
Location
Item
Pieces
SG01
0005
05
SG02
0089
07
SG03
0195
09
SG04
0250
10
SG05
995
12
SHIPPING DETAILS
ORDER DETAILS
ORDER
7 5 16061994 88 00 4
TRACKING
VIEW ITEM

IT systems are a key feature in many aspects of our lives and have an important role in many of the products and service that we encounter on a daily basis. Organisations, in many different sectors, use IT systems to manage different aspects of their business, including monitoring production, improving productivity and supporting customers. In this topic, you will explore how IT systems are used in organisations to support a range of business operations so they can meet business and customer needs.

TOPIC 14 USING IT SYSTEMS IN ORGANISATIONS

LEARNING OBJECTIVES

By the end of this topic you should be able to understand:

- the role of IT systems in organisations
- the concept of transaction processing and how it is used in organisations
- how and why organisations use customer relationship management (CRM) systems
- how management information systems (MIS) are used in organisations
- the concept and use of intelligent transportation systems (ITS)
- operational issues faced by organisations and the impact on IT systems and users.

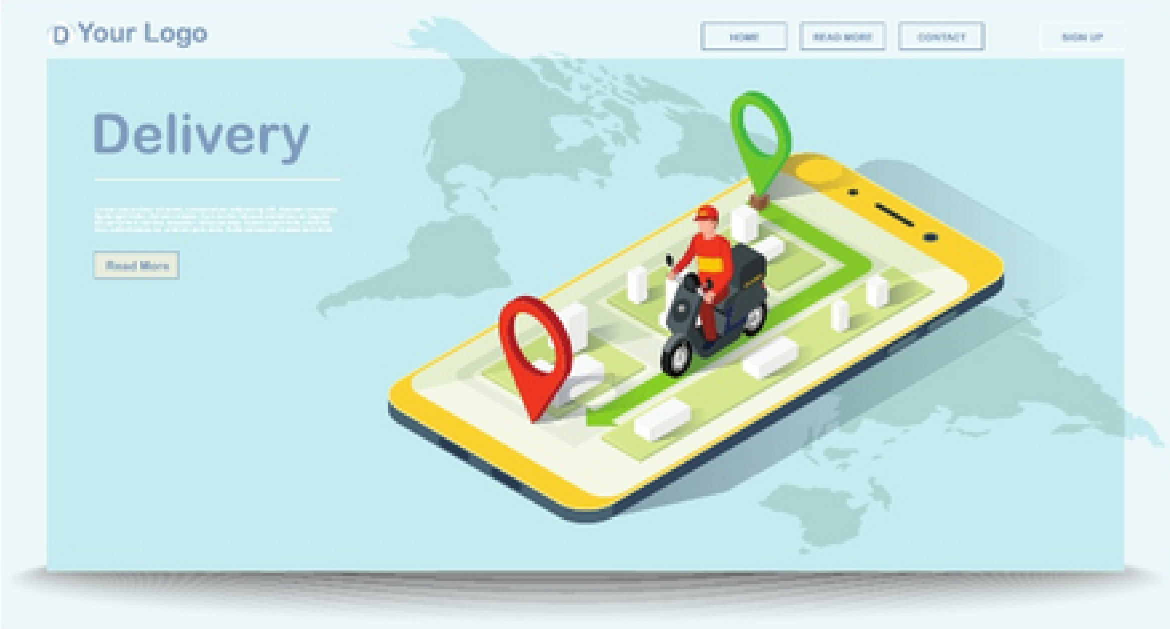
GETTING STARTED

With a partner investigate what is meant by these sectors:

- retail
- education
- manufacturing
- finance
- transportation safety
- commerce
- public services?

For each sector you should:

- provide a definition of the sector
- write down an example of a time you have interacted with that sector. What was the reason you made use of the sector? How did IT systems play a role in this experience?



▲ Delivery drivers are an important part of the retail sector

KEY TERM

business-critical: data, information or tasks/operations that are essential for an organisation to continue to operate

THE ROLE OF IT SYSTEMS

IT SYSTEMS IN ORGANISATIONS

IT systems are indispensable for any modern business or organisation. Whether it is monitoring and controlling a production process, or providing workers with access to **business-critical** information to help them do their job, IT systems are a vital tool that all organisations must use to ensure that they remain effective.

Operational support

IT systems play a crucial role in supporting the day-to-day operations of an organisation. They help to ensure that the individual, everyday tasks that have to be carried out by management, employees and clients or customers are completed effectively and without issue.

Effective IT systems will support and enhance these essential tasks so that the main business needs are met, and operational goals such as these can be achieved:

- Improving efficiency and productivity
- Improving decision-making
- Enhancing communication and collaboration
- Reducing costs and increasing profits
- Improving customer service
- Ensuring security and compliance.

To provide effective operational support, IT systems should be applied to many key areas of a business or organisation, as outlined here.

Automation

Automating repetitive, simplistic or laborious tasks allows an organisation to free up employees so that they can be used for far more strategic work, or where a human being will be more appropriate. Tasks such as data entry, report generation, order processing and basic customer service interactions (such as processing a refund) are examples of where automating a process can be more efficient for both the organisation and the customer.

Monitoring and maintenance

IT systems can be used to continuously monitor the performance of other systems and identify variations in performance in order to spot potential issues before they become major problems. This helps to prevent downtime, data loss and disruptions to core operations.

Customer service

IT systems can be used to automate and improve customer service by providing online self-service portals, chatbots and knowledge bases. This enables customers to find answers to common questions and solve basic problems without needing to contact support directly.

Data management and analysis

Organisations can collect and analyse data from various sources to gain valuable insights into their operations. This data can be used to identify areas where improvement is needed or where processes can be optimised, leading to reduced costs or better services and products.

Resource management

IT systems help organisations to effectively manage resources, such as human resources, finances and physical assets. Enterprise resource planning (ERP) systems help organisations combine various areas such as finance, HR and supply chain management into a single centralised computer system.

Collaboration

IT systems provide organisations and individuals with access to tools and resources to enable communication and collaboration among employees, teams and departments. IT systems, including cloud computing platforms, allow teams in different geographical areas to interact more easily. Used effectively, collaboration tools can enhance teamwork and productivity for distributed teams and enable a more flexible approach to work, such as enabling employees to work from home. Some of the ways that IT systems can support collaboration include those listed here.

Communication

IT systems can provide access to communication tools such as email, instant messaging and video conferencing. These tools enable employees to communicate in flexible ways that best meet their needs, no matter where in the world each team member is located.

Collaboration platforms

Different platforms such as Google Workspace, Microsoft Office 365 and Dropbox allow users to store, share and collaborate on documents, spreadsheets, presentations and other files in real time. As well as allowing users to send files and documents to each other, they also allow users to work on the same document simultaneously, promoting effective teamwork and ensuring effective **version control** of files and documents.

KEY TERMS

version control: tracking and managing changes in documents or software

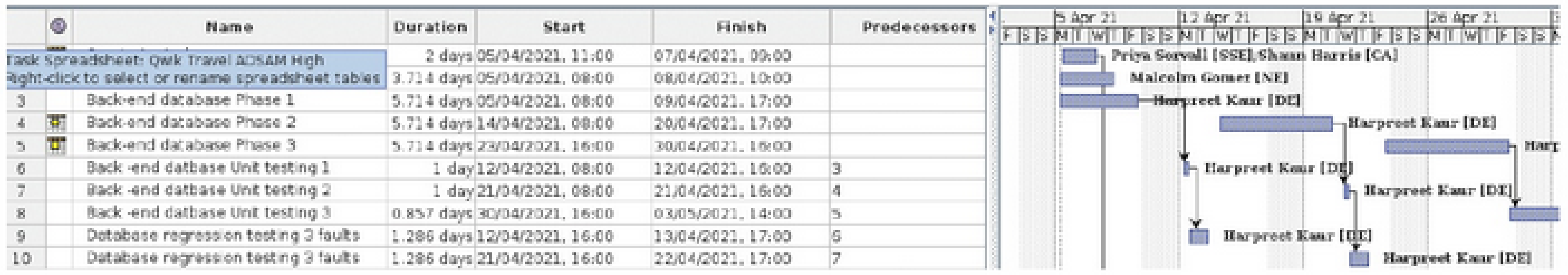
workflow: a sequence of tasks or processes carried out by individuals, groups or organisations to complete an objective

Social media and networks

As organisations become more geographically distributed, greater use is being made of social media platforms and enterprise social networks to promote collaboration and networking among employees. These functions are often provided as extensions of collaboration platforms that an organisation may already use (e.g. Microsoft Teams); at other times they are standalone platforms provided by social media companies such as Meta (Facebook). These tools are often designed to facilitate informal communication, idea sharing and community-building across departments and hierarchical levels in order to improve levels of collaboration and spark innovation.

Project management software

Project management software helps teams to plan, organise and track their work. As well as allowing managers to create long-term plans for whole projects and teams, these tools allow users to plan and manage individual **workflows** for themselves or for other members of a team that they manage. Project management software allows users to set deadlines, assign responsibilities and monitor progress of individuals and tasks, which helps projects stay on track (see Figure 14.1).



		Name	Duration	Start	Finish	Predecessors
		Task Spreadsheet: Qwk Travel ADSAM High	2 days	05/04/2021, 11:00	07/04/2021, 09:00	
		Right-click to select or rename spreadsheet tables	3.714 days	05/04/2021, 08:00	08/04/2021, 10:00	
3		Back-end database Phase 1	5.714 days	05/04/2021, 08:00	09/04/2021, 17:00	
4		Back-end database Phase 2	5.714 days	14/04/2021, 08:00	20/04/2021, 17:00	
5		Back-end database Phase 3	5.714 days	23/04/2021, 16:00	30/04/2021, 16:00	
6		Back -end datbase Unit testing 1	1 day	12/04/2021, 08:00	12/04/2021, 16:00	3
7		Back -end datbase Unit testing 2	1 day	21/04/2021, 08:00	21/04/2021, 16:00	4
8		Back -end datbase Unit testing 3	0.857 days	30/04/2021, 16:00	03/05/2021, 14:00	5
9		Database regression testing 3 faults	1.286 days	12/04/2021, 16:00	13/04/2021, 17:00	6
10		Database regression testing 3 faults	1.286 days	21/04/2021, 16:00	22/04/2021, 17:00	7

▲ Figure 14.1 Project management software allows managers to plan tasks and assign workers

SKILLS

COLLABORATION,
DECISION-MAKING

ACTIVITY 1

Fatimah runs a growing IT training company that provides training and tuition for businesses and individuals.

The company provides a number of training and tuition services including:

- exam tutoring for individuals
- group training for companies on a range of hardware and software.

Training and tutoring sessions can either be face to face at the client's premises or remote via video conferencing software.

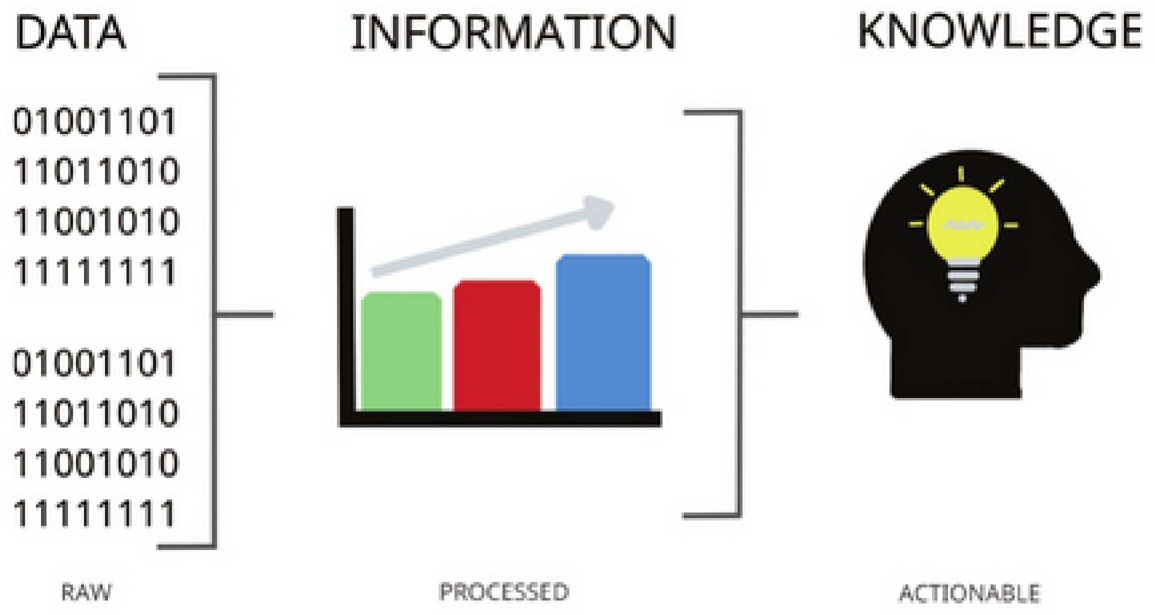
The company employs ten tutors. It currently has an office in a large city centre but, due to escalating costs, Fatimah plans to close the office and change to a fully remote working model.

The tutors plan, create and share tutoring resources with each other and with clients.

Develop a plan for how Fatimah and her staff could make use of the tools and features provided by collaboration software to support key tasks of the company

Knowledge management

As you have seen, IT systems are based on data. Organisations often rely on vast amounts of data (see Topic 12 Manipulating data), which can in turn be processed to produce information. Knowledge management is an extension of this in that it is the process of taking information and organising, processing and manipulating it so that it can be used effectively.



▶ Figure 14.2 Data into knowledge. Can you identify examples of each?

Here are some of the ways that IT systems can be used to facilitate and support knowledge management.

Document management systems (DMS): these enable organisations to store, organise and retrieve documents and files efficiently. DMS platforms offer features such as version control, document tagging, full-text search and access control, so that employees can find and share specific, relevant documents and information.

Content management systems (CMS): CMS platforms enable organisations to create, manage and publish digital content such as articles, blogs, wikis and multimedia resources. These are often used in conjunction with a web platform, such as the organisation's website, and help support collaborative authoring, content editing and versioning. They can be used to support the creation and dissemination of knowledge across teams and departments.

Knowledge repositories: these are used for storing knowledge related to key areas of an organisation's operations. Knowledge repositories will typically include documents, reports, manuals and company policy documents. These are typically used to allow an organisation to centralise important information such as best practice advice or standard operating procedures (SOPs), so that staff can easily access it and ensure they have accurate and up-to-date information whenever they need it.

Product development

Whether it is generating innovative products or refining and improving what an organisation currently offers its customers, product development is vital to ensure a company succeeds. For any company it is important that they remain competitive with rivals, sustain current success, and where possible, grow and expand. Product development is a multistage process that, in modern organisations, is greatly affected by the use of IT systems. The main stages of product development are outlined here.

Ideation, conceptualisation and planning

In the first instance, an organisation will need to identify an initial idea of what is to be developed. This may be a completely new product or just a single feature that could be added to an existing product. Ideas at this stage may go through several phases of refinement as the idea is explored with different stakeholders. IT systems will support organisations as they carry out feasibility studies, engage with stakeholders and record and refine ideas.

Design and prototyping

Once an initial idea has been explored and is considered a viable suggestion for further development, a company will start to produce initial designs for the product. At this stage the company will begin to explore specific details of a product and, depending on the product, the designs can take many forms. For example, a company developing a new smartphone app would produce designs for how it will look – designs for specific functions including flowcharts and code designs. There will also be functional and non-functional specifications. As with the idea stage, it is common to produce multiple versions of designs and go through several iterations of designs and prototypes before the final product design is decided.

▶ IT systems including design software and 3D printing allow for rapid iteration of design ideas

THINKING LIKE AN IT SPECIALIST

Functional requirements define what an IT system will be able to do. A functional specification will include details of the specific features and functions of the system that the user will interact with.

Non-functional requirements describe more general properties of an IT system. A functional specification will include details of how a system will operate in order to support delivery of the functional requirements.

Find and explore examples of functional and non-functional specifications. What are the key differences between the functional and non-functional requirements? Why is it important to define both sets of requirements?

Development and testing

Once a final design has been approved, a product will go into development. Throughout the development of a product there will be multiple testing phases, and again a range of different stakeholders will be involved. It is important to ensure that a product is thoroughly tested during development and, depending on the type of product, this testing can take many forms. No matter what the product, however, it is important to realise that some issues can go unnoticed during the design stage, so ensuring potential users/customers provide feedback early in development can save a lot of time and money compared with discovering an issue after the product is released to the general public.

Production and manufacturing

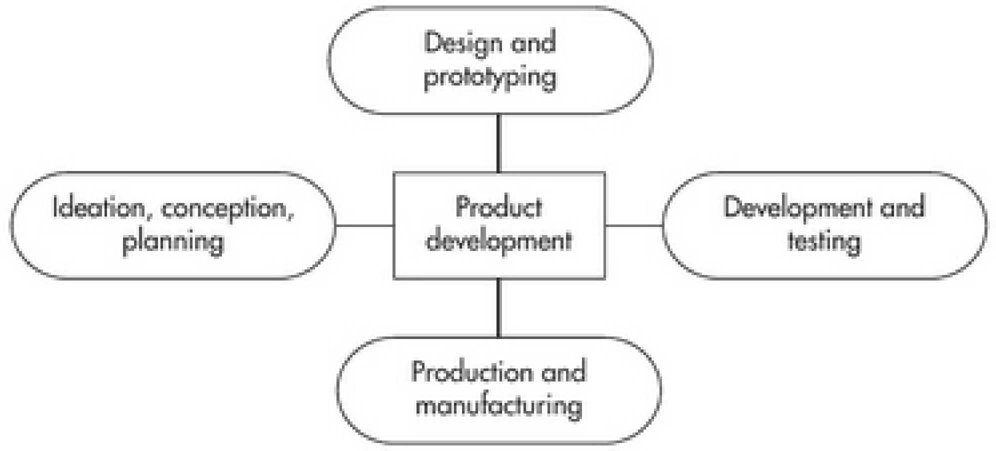
For a physical product it is important that the manufacturing process is both efficient and consistent. IT systems can ensure that the product is produced in the same way every time, which is important for ensuring reliability and predictability in a product.

SKILLS

INTERPRETATION

ACTIVITY 2

Copy and expand this knowledge organiser:



For each of the areas of product development, identify IT systems and digital tools that can be used to support the tasks that an organisation would carry out. For each system provide information on how they would be used.

THINKING LIKE AN IT SPECIALIST

What is the difference between a product and a service?

A product is defined as tangible item. Traditionally this is thought of as a physical item such as an item of clothing or a car. However, with modern systems this can also apply to a digital product such as a computer program or smartphone app.

A service is the exchange of non-tangible goods, such as providing access to specific skills, help, information or even entertainment.

Think of some famous companies/brands. Do they provide a product or a service?

KEY TERMS

back-office processes: administrative and logistical tasks that support a business but do not interact directly with the customer/client

process automation: the use of technology to automate complex, repetitive or mundane tasks

Service delivery

IT systems provide a platform from which an organisation can deliver their services. Using IT systems allows an organisation to streamline how services are delivered, and ensure they reach a wider audience. As with other areas of running an organisation, IT systems can enhance service delivery through **process automation**, effective data and knowledge management, and efficient delivery and optimisation of workflow.

TRANSACTION PROCESSING

Transaction processing, often abbreviated as TP, refers to the management of individual tasks that involve data changes in a system. This is usually for financial or produce-related transactions (such as stock or inventory control). The goal of TP is to maintain data integrity, so the tasks involved are usually short-lived and self-contained events, but they interact with a larger system such as a database. For example, when a purchase is made on an e-commerce site, the item is removed from available stock so that other customers get correct and up-to-date information about availability.

Electronic point of sale (EPOS)

Many modern stores, cafes and restaurants now use electronic point of sale (EPOS) systems instead of the more traditional cash register. An EPOS system provides a range of features that help support effective business practices.

An EPOS system provides a complete solution that allows an organisation to manage sales and inventory, by linking the actions performed by customers and salespeople with the **back-office processes**. Some of the key functionalities include the following:

Sales processing: EPOS systems allow the customer to pay for goods. They typically consist of a cash drawer and a contactless reader to allow customers to pay by card or smartphone.

Inventory management: EPOS systems are linked to the stock database to provide real-time tracking of stock levels. When a sale is made, the quantity held in stock is reduced. If stock levels fall below a predetermined level, items can be reordered automatically.

Customer insights: EPOS systems collect detailed sales data and can identify popular and unpopular items. They can also track days and times when goods sell.

Improved customer experience: EPOS systems can enhance the experience of customers. They are used in self-service checkouts, and with loyalty cards.

SKILLS

COMMUNICATION, COLLABORATION

ACTIVITY 3

The use of self-service checkouts has grown dramatically in recent years. With a partner, discuss the benefits and drawbacks of self-service checkouts for organisations and their employees.

Do you think all customers see self-service checkouts as a benefit?

Order processing

Order processing is a widely used forms of transaction processing. This might have been making It is used when you make a purchase on an e-commerce site or when you buy a ticket at a cinema. Order processing transactions must follow the following ACID principles:

Atomicity: this means that a transaction can't be split – it must be treated as one indivisible unit.

Consistency: the system moves from one valid state to another. So, for example, it may check that the payment method is valid before continuing, or it makes sure that the inventory/stock level is adjusted based on the order.

Isolation: if two customers are buying the same item, the orders should not interfere with each other. In e-commerce this is often done by ensuring minimum stock levels are maintained. Where there are a finite number of transactions, such as booking a seat in a theatre, processes are needed.

Durability: once confirmed, the order details and changes are permanent and are stored securely, even if there is a system failure. Customers and the organisation must be able to rely on the accuracy and availability of the order information after unexpected events.

Financial

Financial systems rely heavily on the principles of transaction processing. Because of the nature of the data being processed, the integrity and security of the transaction is vital. In recent years there has been a significant increase in the number of people globally using digital payment methods instead of cash, so it is important that customers can rely on the system to function whenever they need to make a purchase, either online or in person at an EPOS system. Some of the key considerations for financial transaction processing include the following:

ACID compliance: transactions should follow ACID principles (Atomicity, Consistency, Isolation and Durability) to ensure data integrity and accuracy.

High availability: customers and organisations may need access to their money or payment system at any time of the day, so systems must be available 24/7, and have enough redundancy built in that there is no system downtime from the users' point of view.

Scalability: systems must be able to handle large volumes of transactions efficiently and should be able to respond to spikes in access requests; for example, an increase in sales of gifts before significant holidays or celebrations.

Security and fraud prevention: robust measures must be in place to protect sensitive financial data and prevent unauthorised access or fraudulent activity.

Compliance with regulations: the financial sector is subject to regulations that are designed to ensure transparency and reduce financial crime. Systems must follow these guidelines, which places an additional burden on the system designers and the organisations using them.

Bacs Payment Schemes Limited (Bacs)

Bacs Payment Schemes Limited (Bacs) is an organisation based in the United Kingdom that is responsible for the operation of several major payment systems, primarily focusing on direct debit and direct credit transactions.

Direct debits are payments that a customer agrees to allow a company to take money from their bank account on a regular basis, such as for a utility bill or phone contract. Direct credits are payments that are made directly into a bank account, such as for a salary or government benefit.

Although Bacs only operates in the UK, there are similar organisations in other countries, for example Automated Clearing House (ACH) in the United States of America, and SEPA (Single Euro Payments Area) in the European Union.

SKILLS

INTELLECTUAL INTEREST AND CURIOSITY, REASONING

ACTIVITY 4

If you are outside the UK, research which organisation performs a similar function to Bacs in your country. How is it similar to Bacs and how does it differ?

If you are in the UK, explore why Bacs only operates within a single country. What prevents the organisation from working in global financial markets?

Produce a summary of your findings and compare them with another member of your group.

CUSTOMER RELATIONSHIP MANAGEMENT SYSTEMS

Customer relationship management (CRM) systems are software tools designed to help businesses manage interactions and relationships with their customers and other stakeholders. CRM systems typically make use of database tools along with other functions to allow them to provide targeted support and actions on a range of other platforms. As the name implies, CRM systems aim to help companies provide a positive experience for stakeholders in order to increase customer engagement and ultimately increase profits.

Synchronise marketing events

Many CRM systems provide tools to enable companies to ensure that they are using appropriate marketing across a range of platforms. CRM systems allow companies to plan a series of related campaign activities, such as social media posts, emails or SMS messages, and schedule them from within the system. CRM systems will also often have tools that integrate with other platforms (such as social media) so that advertisements and messages can be sent out automatically across all platforms to ensure that the same message is being presented across different mediums.

Loyalty schemes

A loyalty scheme is a marketing and customer relations tool used by companies to incentivise customers. CRM systems are often closely linked to EPOS systems and inventory databases, and use data analysis to identify trends and patterns on a company-wide basis, and also at an individual customer level. A loyalty scheme will provide a general incentive for every customer – for example, the customer may earn points based on the value of their transaction. These points can often be saved and used for rewards at a later date, such as money off a future purchase or special prizes. Effective loyalty schemes, however, make use of CRM to personalise offers and additional incentives. For example, the CRM system could identify that a customer has previously purchased a particular brand of breakfast cereal. The customer would then be sent a voucher or special offer for that brand of breakfast cereal to encourage them to continue using their store to purchase that product.

Buying trends

CRM systems can provide companies with valuable insights into the sales data generated by their systems. These insights allow the company to make strategic decisions, including when and how much stock of a particular item should be purchased, when advertising a particular product would be most effective and to whom the advertisements should be targeted.

Customer service

CRM systems allow companies to personalise a customer's experience to make them feel more valued. For example, a company might send out a personalised greeting or special offer on a customer's birthday, or send a follow-up email about a purchase that they made. CRM systems also allow companies to track issues that have been raised by customers, allowing them to effectively follow up on enquiries or ensure problems are resolved in a way that is satisfactory to the customer.

Customer retention

Customer retention is a key factor in the success of a business. Attracting a customer for just a single purchase is the sign that a previous advertisement campaign or deal has worked effectively. However, managing to secure a return customer means that the customer is happy with a much wider scope of the business. Returning customers are also more likely to recommend a company to others, so securing return custom is vital to business success. CRM systems allow companies to draw data, information and knowledge from all aspects of the business to provide support for customers and incentivise return custom.

▲ CRM systems allow companies to target specific groups or individuals with relevant information

Upselling

CRM systems provide insights into the data that they hold to allow them to identify trends and patterns, which can be used to improve sales and increase profit. For example, a user goes to an e-commerce site to buy a new laptop. They search for the specific laptop they want and add it to their shopping basket. The CRM system, which is integrated with the e-commerce store, would then compare this laptop to similar purchases made by other customers. Based on this it may recommend other related products as 'add-ons' to the order, such as a carrying case, or peripherals such as a mouse or headset. These recommendations often encourage impulse purchases and help increase company profit.

MANAGEMENT INFORMATION SYSTEMS

Management information systems (MIS) are systems and processes that support the operations of an organisation by providing managers with the information they need to effectively run the organisation and make data-driven decisions. MIS encompass a range of technologies, processes and methodologies designed to collect, process, store and disseminate information. Their primary function is to draw together data and information from a range of sources and combine it into a single usable system.

Record keeping

MIS gather data from various sources appropriate to the organisation. Internal sources may include transactional systems (e.g. EPOS, CRM), employee databases, financial records and operational systems. They then combine this with data from external sources, which may include market research, industry reports, government databases and competitor analysis.

As with any data system, the data must be processed in order to be useful; the MIS will apply data-processing tasks such as validation, cleansing and aggregation (see Topic 12 for detailed information on data-processing tasks) to prepare the data for use.

Depending on the data involved, and the needs of the organisation, the MIS will work with structured, semi-structured and unstructured data. The job of the MIS is to remove the complexity of the data and present it to the user in an accessible way.

Decision-making

Effective decision-making is supported through high-quality data that gives detailed insights that can be used as actionable knowledge. An effective MIS should provide users with mechanisms for retrieving information in an effective and efficient manner. Common tools include the use of query tools, search engines, dashboards and reporting interfaces that allow users to access relevant information based on their needs and preferences. Information retrieval systems are designed to provide users with quick access to up-to-date and accurate data to support decision-making.

CASE STUDY: ESHO SHOES

The owner of Esho shoes is using a management information system (MIS) to analyse data to identify opportunities for efficiency improvements.

Analysis of production downtime data provided by the MIS shows that a particular manufacturing line experiences frequent stoppages during shift changes.

Explanation

Without the MIS the manager may have only the production figures to use to make decisions, which would show a reduction in output, but would not provide contextual information for them to effectively diagnose the cause.

The MIS draws together data on equipment downtime, production delays and operational inefficiencies, and is able to highlight a recurring pattern, that the downtime occurs during shift changes.

Implications

The insight suggests that the transition between shifts is not optimised, leading to inefficiencies and reduced productivity. During shift changes, there may be delays in equipment set-up, coordination issues between incoming and outgoing shift workers, or other factors contributing to downtime.

Actions

Based on the insight from this data, the manager was able to implement specific actions to minimise down time on this production line, including:

- optimisation of shift-change procedures to improve communication between teams on different shifts
- additional training for employees so they were able to perform multiple tasks and roles, ensuring smoother transitions between shifts and reducing dependency on specific individuals for critical tasks
- changes in the maintenance schedules to minimise unplanned downtime.

Summary of impact

Because the MIS was able to provide a much more specific view of the problem, in particular that the issue was being caused by a specific event (shift changeover), the manager was able to look specifically at potential root causes for why the identified issue was occurring.

As a result, the manager was able to identify potential solutions and implement them more quickly, saving both time and money for the company.

Project management

As you learned earlier in this topic, project management software plays a key role in supporting employees who work in a collaborative environment. MIS play a crucial role in supporting project management by providing tools and capabilities that can be integrated with and used alongside project management software. By using MIS, an organisation can identify key insights with data-driven decisions about the focus and direction of a project. For example, it may identify an issue in a product that leads to the development of new and more efficient manufacturing process, or during a project it may identify potential risks to a project so that these can be mitigated more effectively, or more effective contingency planning can be established.

SKILLS
ADAPTIVE LEARNING, ANALYSIS

ACTIVITY 5

Marius owns five grocery stores across a large city. The stores sell a range of goods including fresh fruit and vegetables.

The chain of stores uses an EPOS system that provides Marius with data about each of the stores and the company as a whole.

Marius has noticed that one of the stores has a much higher rate of wasted fresh produce, when compared with the other stores.

Analyse how Marius could use the EPOS system and the data that it provides to discover why one particular store wastes more fresh produce than other stores.

INTELLIGENT TRANSPORTATION SYSTEMS

Intelligent transportation systems (ITS) aim to improve the planning and operation of public and private transportation. ITS uses various sensors, cameras and traffic monitoring systems to collect real-time data on traffic conditions, including congestion, accidents, road closures and weather conditions. By analysing this data, ITS can identify traffic bottlenecks and levels of demand, and predict potential issues. These insights then provide actionable knowledge that can be used to plan improvements or make dynamically real-time adjustments.

Scheduling and route planning

One aspect of ITS is the use of advanced routing algorithms that take into account real-time traffic information, road conditions and other factors to calculate optimal routes for vehicles. These algorithms consider factors such as traffic flow, road capacity, speed limits and vehicle restrictions to determine the most efficient path for each vehicle, reducing travel time and fuel consumption. These systems can be highly useful for logistics or on-demand transport such as taxis.

ITS can also be used for scheduled public transport that often has a predetermined route. Uses of ITS in public transport can include the following:

Dynamic scheduling: bus and train schedules are adjusted based on actual passenger numbers and traffic congestion, reducing wait times and optimising service efficiency.

Demand-responsive scheduling: flexible routes are generated based on real-time demand signals from passengers; times and routes can be adjusted to avoid areas and times where no passengers will be waiting and redirect them to areas where passengers require transport.

Incident management: schedules can be quickly adjusted around accidents or unexpected events to minimise disruptions.

Timetabling

The algorithms in the systems can be used to predict future patterns and demand. For example, using historical data and machine learning, the system may identify specific areas of a city with high demand at the beginning or end of the working day, so more buses at more regular intervals are required in those areas at those times. It may also anticipate future demand, such as a steady increase in use of a specific route, which may suggest that this demand will continue to increase in the future.

Locations

An ITS makes use of location-based services to collect and analyse location data to generate insights into travel patterns, traffic flows and user behaviour. This data can be collected in many different ways. For example:

User-generated data: this is one of the primary sources of data for an ITS. Location-based services that work in conjunction with mapping apps on smart devices provide the ITS with anonymised speed and location data. This data is aggregated to create a real-time picture of traffic flow in specific locations. The more users who have enabled location services, the more accurate and detailed the traffic information becomes.

Community-based reporting: traffic reporting from members of the public and broadcast services such as radio stations contributes valuable real-time updates on accidents, road closures and other incidents in specific locations.

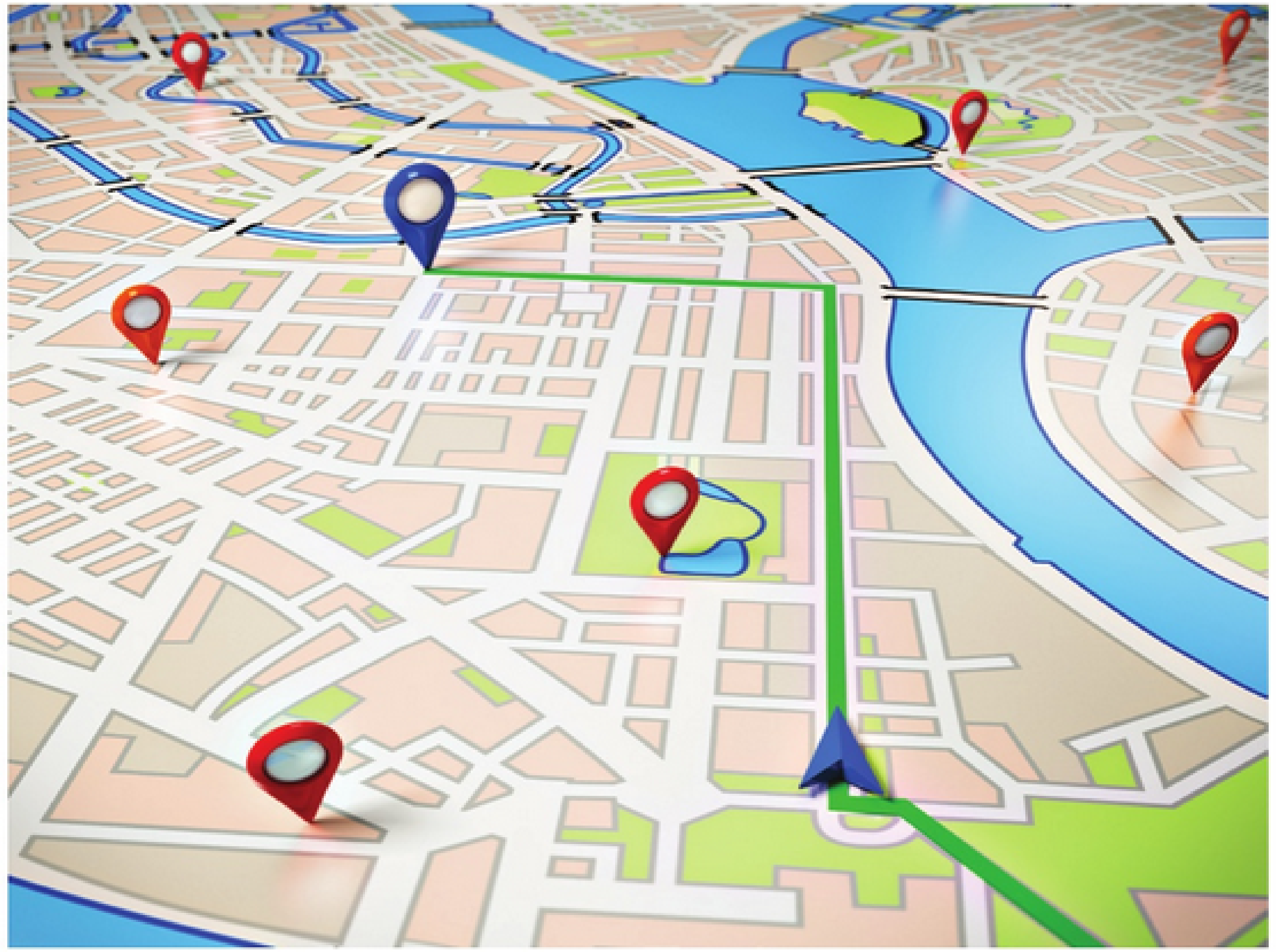
Road sensors and traffic cameras: ITS works in collaboration with government agencies and private companies to access data from traffic sensors and cameras.

Fleet management

KEY TERM

telematics: an in-vehicle system that uses GPS and vehicle diagnostics to provide detailed, real-time information

An ITS enables real-time tracking and monitoring of vehicles using GPS and **telematics** technology. By tracking vehicle locations, speed and status in real time, an ITS allows logistics companies to track vehicles and monitor their status. As well as speed and location, these systems can provide managers with information about how the driver is controlling the vehicle, for example, sudden braking or making erratic or violent turns. This can be especially useful for fleets that deliver fragile cargo. The systems also allow fleet managers to oversee operations to enable proactive scheduling and dispatching of vehicles to optimise routes and respond to changing demand.



▶ Location-based services allow fleet managers to track vehicles in real time

EXPERT SYSTEMS

An expert system is an IT system that helps support decision-making in a specific area of knowledge, such as medicine or technical troubleshooting. An expert system is programmed with a knowledge domain and a series of rules and reasoning mechanisms that allow it to solve problems or make decisions in that domain.

An expert system typically has a narrow scope; that is to say, it will usually work within very strict parameters. For example, in the medical field a system may focus only on suggesting prescription medication based on specific symptoms. Although linked to some degree, expert systems differ from machine learning and artificial intelligence in some key areas.

Knowledge-based: expert systems rely on pre-programmed rules and facts encoded by human experts in a specific domain, rather than being trained on large data sets and using an algorithm to 'learn'.

Static: their knowledge base doesn't adapt or improve automatically through new data.

An expert system is made up of two main elements, the **knowledge base** and the **inference engine**. The knowledge base acts as the brain of the system; it is a repository of a vast amount of information related to the specific area of expertise of the system. This repository will hold facts, rules and heuristics (strategies based on the experience of experts).

The inference engine is the decision-making part of the system. It uses the information extracted from the knowledge base to process issues, draw conclusions and recommend solutions to a given problem. In simple terms, the knowledge base allows a problem to be diagnosed; the inference engine identifies possible solutions.

KEY TERMS

inference engine: the decision-making part of an expert system

knowledge base: a large repository of information related to a specific area of expertise

ACTIVITY 6

A software company has an online help system that uses a chat screen where customers can ask a technician for help by typing and reading messages.

The company wants to change the system so that the customers' initial contact is with a chatbot powered by an expert system. The chatbot would let the customer communicate with the expert system using a chat screen, and the expert system would then try to diagnose the problem and suggest a solution. A human would only get involved if the system was unable to solve the problem.

1. Draw a dataflow diagram (you learned about these in Topic 4) for the new system.
2. What challenges do you think the company will have in developing the system?
3. What are the benefits and drawbacks for the company of using such a system?
4. What are the benefits and drawbacks for the customers of using such a system?

OPERATIONAL ISSUES

IT GOVERNANCE AND POLICY

IT governance typically refers to the pre-defined processes, structures and mechanisms in an organisation that aim to make sure IT resources are used appropriately and effectively. As with many areas of organisations, IT governance exists to allow the organisation to achieve its key business goals by ensuring the organisation manages and mitigates risks, and complies with legal and regulatory requirements.

In any organisation, effective governance and policy documents should clearly define rules and procedures for all employees in terms of their use of IT systems. The documents will also define specific roles and responsibilities of key personnel, particularly those with responsibility for the care and upkeep of the IT systems. Policies should also clearly define decision-making hierarchies so that ultimate responsibility for business-critical decisions can be identified. For example, which member of staff would make the final decision to replace the central server, and when that should happen.

The key goals for IT governance include the following:

Strategic alignment: IT activities and investments should be in line with the organisation's overall business strategies and goals.

Risk management: potential risks to IT systems should be identified along with ways to mitigate them so that systems are protected, and potential issues can be avoided.

Resource management: the purchase, use and allocation of resources should be carefully planned to maximise value and minimise waste.

Performance measurement: key success and performance measures should be established to assess the effectiveness of IT governance processes and make data-driven decisions to improve performance.

Compliance and legal requirements: companies must ensure that they comply with all relevant protection regulations, privacy laws, cybersecurity requirements and other legal obligations. Failing to comply can result in legal action that can be financially damaging or lead to a loss of trust from customers.

In order to achieve IT governance goals, IT policy will often focus on the following key areas.

Business continuity

It is vital for an organisation that in the event of a serious data breach or system failure they can continue to perform their core functions. Business continuity is essential for maintaining customer confidence, minimising any potential disruption for or impact on customers, and minimising the financial impact on the organisation. The business continuity plan will define specific procedures on how to respond to issues, including what fallback systems and process should be used, and who is responsible for key actions in response to an identified issue. Key actions and roles may include informing customers and regulatory bodies, investigating root causes, mitigating impact and reviewing and updating current procedures. The business continuity plan is often linked to or integrated with the disaster recovery plan.

Disaster recovery

In the event of a serious issue, such as a critical system failure or data loss, it is vital that a company's data is restored as quickly and completely as possible. The disaster recovery policy outlines the organisation's procedures for responding to IT disasters. Disasters can include system failures, natural disasters or cyber attacks. An effective policy will clearly specify recovery objectives, roles and responsibilities of key personnel, backup strategies and contingency plans to minimise downtime.

Risk management

Risks cannot always be avoided in a business environment, so it is vital that a company thoroughly identifies potential risks, and plans for ways that these can be reduced and mitigated. For example, as a company grows and is more visible, the risk of a cyber-attack also increases. It is not possible for the company to prevent a malicious attack, but it can reduce the risk presented by an attack by making its systems more resilient and having appropriate data protection and recovery procedures in place. As with other policies, a risk management policy should not be considered in isolation; it should be cross-referenced with other relevant policies such as the disaster recovery policy and the security policy.

SKILLS

ANALYSIS, CRITICAL THINKING

ACTIVITY 7

Investigate common risk assessment methodologies used to assess potential risk to computer systems.

Use what you have learned to carry out a risk assessment of the IT systems in your school or college.

You could copy and use this basic risk assessment table to structure your work, or you could use a different layout that better suits your chosen methodology.

Potential risk	Probability	Severity	Risk score	Comments and potential mitigation

User policy

A user policy, also often referred to as an acceptable use policy (AUP), defines acceptable behaviours and practices for using IT resources, including computers, networks and software applications. This policy will typically list what a user can and cannot do when using or accessing company IT resources, and it will provide guidelines for employee conduct, internet usage, email communication and the handling of sensitive information. The AUP will often define the potential consequences for failure to follow the AUP. It helps to support good system security, but it is also useful to define the professional expectations of employees in terms of how they conduct themselves in their work, and when communicating or interacting with both internal and external stakeholders.

MANAGING IT CHANGEOVER

It is important for an organisation to continually review its systems and processes. When a new system has been identified and developed it is important to the success of the new system, and the success of the business, that the transition between the old system and the new system is managed appropriately. There are a number of ways that a changeover from one system to another can be managed, and each method has its own benefits and drawbacks. These benefits and drawbacks should be carefully considered in relation to the specific system, organisation and context, as each method will be suitable for different scenarios.

Phased changeover

A phased changeover involves gradually implementing the new system or process in stages, while maintaining the existing system until the transition is complete. This method is most suitable when the new system can be easily divided into distinct components or modules. For example, in an enterprise resource planning (ERP) system for a large manufacturing company, the system would be gradually developed to cover modules related to different parts of the business. This changeover could look like this:

Phase 1 – Finance and accounting

Phase 2 – Inventory and procurement

Phase 3 – Sales and order management

Phase 4 – Reporting and analytics.

The phased changeover method allows an organisation to reduce risks as each phase is tested and refined before the next phase is implemented. This means that any issues that are discovered are restricted to a single area of the business, and some of these issues can be anticipated and avoided during the next phase. Slowly introducing the system also gives users time to adapt to the new system – since it is introduced gradually, users can be trained to use the system in smaller groups, which reduces the burden on the organisation.

However, the gradual nature of the changeover means that it takes longer for the organisation to see the full benefits of the system. The organisation also needs to consider the compatibility between the old system and new system. For example, if the finance and accounting department has the new system and the other departments do not, how will the data and information produced by the new system be transferred into and used by the old system?

Direct changeover

Direct changeover, sometimes also referred to as 'Big Bang' changeover, involves replacing the entire old system with the new one in one go, usually during a specific cut-off period. For example, a company may shut one system down at close of business on a Friday evening, then resume work on Monday morning using the new system.

The sudden changeover means that the organisation will start to see the complete benefits of the new system immediately. It also reduces the complexity of the changeover as there is no need to plan for how both systems can be used at the same time, and the issues that could occur because of this.

However, this method can be risky as it does not provide a fallback for the organisation to use should there be an unforeseen issue with the new system. With direct changeover, it is important that all personnel have been fully trained in the new system before the transition is made. Even then, it can result in an initial drop in productivity as users get used to the new system. Because of this, direct changeover is only recommended for small-scale or non-business-critical applications; for example, changing the office productivity suite.

Parallel changeover

Parallel changeover involves running both systems simultaneously for a set period of time. A small amount of the workload is carried out on the new system and the rest is carried out on the old system. Over time, more and more of the workload will be transferred to the new system until the new system is doing all tasks and the old system can be shut down and removed. It is also common in parallel changeover to duplicate some of the workload across both systems so that the new system can be tested and compared with the old system, then refined as needed.

The parallel changeover reduces potential risks caused by system downtime as there is a fully working fallback available throughout the transition period. However, it may increase the time spent on transition, which increases the resources (physical and human) required, so increasing costs.

Pilot changeover

A pilot changeover implements a full and complete system, but within a limited area of the business or with a particular group. The identified area of business will use only the new system for a specific amount of time, during which the system will be tested, reviewed and refined. At the end of the pilot, the organisation will evaluate the results and decide if the system is to be adopted by the business.

The pilot changeover allows a system to be tested in a real-world setting, so a true idea of how the system performs can be established. The risks to the organisation are restricted to only the pilot area, meaning that if there is a significant issue or critical failure of the new system, the overall impact on the business is minimised.

Because the pilot is restricted to a specific area or group, it may not fully reflect how it will perform in other areas of the organisation, where challenges may vary. The pilot method may also not identify issues with using the system on a large scale. It is often sensible to combine the pilot methodology with another methodology when rolling the system out to the rest of the organisation.

CASE STUDY: DRAC AND HASC INVESTMENTS

Drac and Hasc Investments is a wealth management firm that provides investment services for commercial and private clients. The company was looking to improve its client onboarding process.

The old system was paper based, slow and prone to errors, leading to frustrated clients and inefficiencies. They decided to implement a new cloud-based onboarding platform, but recognised the potential risks of a full-scale switchover.

Solution: Pilot methodology with phased approach

Drac and Hasc opted for a pilot methodology with a phased approach to mitigate risk and ensure a smooth transition.

Phase 1: Pilot with a small client segment

- The company selected a group of 50 new clients representing diverse investment profiles across both commercial and private clients.
- The new platform was launched and implemented for use with this pilot group and the relevant staff only.
- The staff responsible for the pilot group were provided with training on the new system and how it should be used with clients.
- Continuous testing was carried out on the system during the pilot phase and feedback gathered from staff and clients.

Outcomes from the pilot:

- Some minor bugs and usability issues were identified in the platform.
- Feedback from the clients and staff provided valuable user preferences.
- Staff training materials and client support resources were updated based on feedback.

Phase 2: Phased rollout to larger client base

- Based on the outcomes of the pilot, the platform was refined and identified issues were addressed.
- The new platform was rolled out to a further 200 new clients in phases based on client type and adviser familiarity.
- The development company provided ongoing support and training throughout the rollout process, continuously monitoring performance and addressing any emerging issues.

Outcomes from the phased rollout:

- Reduced onboarding time by 30 per cent: the new platform streamlined data collection and document management, significantly speeding up the onboarding process.
- Improved client satisfaction: clients appreciated the faster and more user-friendly onboarding experience, leading to higher satisfaction scores.
- Increased efficiency: advisers saved time and effort by using the automated features and centralised data management.
- Reduced errors: the digital platform eliminated manual data entry, minimising errors and ensuring data accuracy.

Summary

The pilot methodology allowed Drac and Hasc to identify and address potential issues before wider implementation, which minimised risk and potential disruption.

The phased approach following the initial pilot ensured a smooth transition for both advisers and clients leading to greater user satisfaction and a greater willingness to use the new platform.

During both phases continuous feedback and iteration were crucial for refining the platform so that when a full changeover had occurred the potential for unforeseen issues had been significantly reduced.

Take it further

Consider other changeover methodologies that Drac and Hasc could have adopted. How might other changeover methodologies have been employed? What would be the benefits and risks in this scenario?

SYSTEM MAINTENANCE

System maintenance is the ongoing process of checking, managing and updating systems to ensure they continue to be effective. It is vital for an organisation to ensure that business goals are effectively met, which means that IT systems should be continually tested and reviewed to ensure they continue to be reliable, as well as continuing to effectively perform necessary tasks.

Perfective maintenance

Perfective maintenance aims to identify and implement potential improvements to already working systems. This form of maintenance is used to increase efficiency, enhance performance or add additional functionality.

Perfective maintenance may be part of the company's ongoing test and review process, or it may occur in response to user feedback, market trends or changes in the company's strategic objectives.

Adaptive maintenance

Adaptive maintenance involves modifying the system to accommodate changes in the external environment. Adaptive maintenance may be triggered by installation of new hardware and software that is incompatible with the old systems, or as a result in a change in legal or regulatory requirements.

Adaptive maintenance often requires changes to back-end processes, system code, configuration or underlying architecture.

Corrective maintenance

As the name suggests, corrective maintenance is the process of fixing errors, bugs or defects in a system. Corrective maintenance is typically reactive in nature and is usually implemented in direct response to a specific identified issue, either through user feedback or because abnormal behaviour is detected in the system.

Corrective maintenance involves analysis of the potential root cause of a detected issue so that a fix can be developed and applied. Following corrective maintenance, you should always perform regression testing – a series of tests to ensure no additional or unforeseen issues have arisen as a direct result of the alterations made to the system.

SKILLS

ADAPTIVE LEARNING, CRITICAL THINKING

ACTIVITY 8

A small group of users of a website have reported that they are unable to log into the system using their usernames and passwords.

1. What type of maintenance will be required to identify and rectify this problem?
2. Describe the processes that the website's developer would follow to fully rectify this issue.

ARCHIVING DATA

Organisations can produce massive amounts of data; while some data is used continuously, some data is used less often the older it gets. This data may need to be moved out of a primary system in order to maintain performance. However, the organisation may still need access to the data, just less frequently. Data archiving is the process of storing historical data in a way that ensures that it can be accessed when needed while remaining secure.

There are a number of reasons why an organisation needs to archive data.

Compliance and regulatory requirements: many industries have requirements that are governed by local laws dictating the need to retain records for specific periods of time, even for customers who no longer use their services. Archiving data helps companies comply with regulations and industry-specific standards, while reducing the strain on primary systems.

Business continuity and disaster recovery: archiving data provides additional backups of critical information, which protects against data loss through system failures, disasters or cyber attacks.

Historical analysis and reporting: archived data can be valuable for historical analysis, trend identification and long-term planning. Archived data can be accessed by management information systems to provide key insights and to inform decision-making.

Auditors and compliance audits: regulatory agencies and external auditors may require organisations to provide evidence of data integrity, security and compliance with industry standards. Archived data can provide a record of past activities, transactions and communication, which can help establish compliance.

Legacy systems and migrations: as organisations upgrade or replace systems, applications or technologies, data may need to be migrated or integrated into new platforms. Archiving the data ensures that legacy information remains accessible during system migrations or upgrades, preserving continuity of operations and data integrity.

DISASTER RECOVERY PLANS

As you learned earlier in this section, a disaster recovery plan is a vitally important part of an organisation's IT governance and policies. A good disaster recovery plan should ensure that in the event of a critical system error, downtime is minimised, data is protected and business continuity is maintained.

Key data

The following must be taken into account.

Inventory of data: a disaster recovery plan should provide a detailed inventory of the key data that is covered by the plan. The plan should identify and describe all critical data and systems that need to be protected.

Recovery objectives: for each of the identified critical data points and systems, the plan should identify objectives for recovery. This will include:

- recovery time objectives (RTOs) – the timeframe in which the data should be recovered and restored. The RTO should be established based on its level of importance in ensuring business continuity.
- recovery point objectives (RPOs) – the maximum amount of data that an organisation is willing to lose in the event of a disaster. For example, an RPO of one hour would suggest that if a system went down the company would expect to lose no more than one hour's worth of work/data – therefore, this data would need to be backed up every hour.

Backup and recovery procedures: for each of the key data points the plan should provide detailed instructions for how the data should be backed up, how the backup should be stored and how it should be protected.

Risk analysis

The risk analysis should analyse specific, potential threats to data and systems. The risk analysis will consider the nature of different threats, the potential impact these could have in relation to the organisation's core functions and ways in which risks can be mitigated. The three key strands of the risk analysis include:

1. Threat identification: a comprehensive assessment of potential threats and risks to data and systems should be carried out by a nominated individual (often the senior IT manager). The assessment should identify specific threats from both internal (e.g. human error, malicious employee) and external (e.g. natural disasters, cyber attacks). It is important that identified threats are specific so that clear mitigation can be planned.
2. Impact assessment: for each identified threat there should be an evaluation of the potential impact it will have on the data, the system and on business operations. This assessment can feed back into other areas of the plan, such as defining RTOs and RPOs based on the assessment of how critical the identified data or system is.
3. Risk mitigation strategies: for each threat there should be a detailed and specific approach to reducing the likelihood of the threat occurring. It is important to realise that risk mitigation does not fully guarantee that the threat will not occur, so planned mitigation should also consider strategies for reducing the potential impact should an issue occur.

Team actions

The disaster recovery plan should clearly define the members and roles of the disaster recovery team. In a large organisation this may well be a dedicated team whose sole responsibility is protecting against data loss. In smaller companies it is likely to be members of the main IT support team.

Defining team roles: the disaster recovery plan should clearly identify which members of the team are responsible for what actions, and the schedule for each of these actions. The plan should also include details of who is responsible for a particular task if the primary delegate is unavailable.

Communication plan: a detailed plan for internal and external communication during a disaster should be included in the plan. It should contain contact information for team members and key stakeholders.

Training and drills: there should be ongoing and regular training for the disaster recovery team and other employees on their roles and responsibilities in a disaster. Conducting regular drills to test the plan's effectiveness will allow the team to identify potential issues and shortfalls, and correct these before an actual crisis occurs. Keeping records of all training and drills along with the disaster recovery plan can be valuable information in the event of an audit.

Management

It is essential that a disaster recovery plan is not a static document. Developing a plan but leaving it unread and unused, can lead to problems should a critical event occur. It is important to manage the plan to ensure that it is used effectively.

Plan approval: once the disaster recovery team have developed the plan it should then receive formal approval from senior management. Senior management would then be expected to commit to ensuring all necessary resources are provided.

Plan maintenance: the plan should be regularly reviewed and updated to reflect potential changes in business operations, or in light of new and emerging threats. Whenever the plan is updated, it should once again be sent to senior management for approval.

Documentation: both the plan and any other related system or technical documentation that would assist in protection procedures and recovery should be easily accessible to all relevant users. Documentation should be clear and concise and stored in a separate system that would be unaffected should a critical event occur.

CHECKPOINT

1. Name the five key areas that IT systems support in an organisation.
2. Explain what is meant by transaction processing and provide an example of how it is used in organisations.
3. Describe how an organisation might use customer relationship management (CRM) systems.
4. Describe how management information systems (MIS) are used in organisations.
5. What are intelligent transportation systems (ITS)? What are some of the key features of these systems?
6. Why is it important for a company to develop policy to manage operational issues and IT systems?

EXAM TIP

If the question contains a scenario, such as a specific company or industry, try to contextualise your answer to that scenario. Learners who apply what they know to the given scenario tend to perform better, particularly in the extended open response questions (discuss, analyse, evaluate).

SKILLS
CRITICAL THINKING, ANALYSIS

EXAM PRACTICE

1. Explain **two** ways that IT systems can provide operational support for an organisation. [4 marks]
2. Name **three** features provided by an online platform that would support collaboration. [3 marks]
3. A small supermarket chain is introducing new electronic point of sales (EPOS) systems. Explain two benefits to the company of introducing EPOS systems. [4 marks]
4. Describe how an expert system differs from an artificial intelligence (AI) system. [4 marks]
5. An order-processing system should make use of ACID principles. Discuss the ways that ACID principles help to ensure the integrity of data in an order-processing system. [6 marks]

TOPIC 15 SYSTEMS DEVELOPMENT

Whether it is the installation of a new network infrastructure in a small city centre office, or developing a new management information system for a multinational corporation, the successful development of any computer system hinges on effective planning and organisation. In this topic you will explore the key knowledge and skills required for managing an IT project, as well as looking at common systems development methodologies that are used to plan, develop and maintain IT systems.

TOPIC 15 SYSTEMS DEVELOPMENT

LEARNING OBJECTIVES

By the end of this topic you should be able to understand:

- the need for project management when developing systems
- the characteristics of a successful IT project
- why project outcomes use SMART targets
- how to specify SMART targets
- the tools that are used to manage projects and how to use them
- the concept and features of the waterfall development methodology
- the concept and features of the agile approach to development methodology
- how to create and interpret information flow diagrams.

GETTING STARTED

Think about a substantial task or project that you have worked on recently where you had to work with other people. This might be a class project, or a task outside of school/college.

- How did you plan and organise the work?
- How did you delegate the work?
- How did you keep track of progress of the tasks and others in the group?
- Was the project successful? What went well and what didn't? Why?

Make notes on how you think the project could be better organised if you were involved in a similar project in the future.

▲ IT projects can be extremely complex. Digital planning tools help to manage and reduce potential complexity

KEY TERMS

feature creep: continual addition of new features to a product in development that affects the team's ability to complete the project effectively and negatively affects deadlines and budgets

mitigation: an action or strategy intended to reduce the seriousness or harm caused by a potential risk

project manager: the individual responsible for overseeing and delivering a project by managing human and physical resources

stakeholder: an individual or group with a particular interest in or link to a project

PROJECT MANAGEMENT

CONCEPT OF PROJECT MANAGEMENT

When developing IT systems, or on any substantial work project, it is important that the project is managed effectively. IT projects have many different aspects that need to be monitored; failing to keep track of one area can have a serious effect on the success of the project in terms of both the final quality and also in meeting any agreed deadlines and costs. IT is a fast-moving field, where there are constantly new technologies and trends emerging. Organisations may want to adopt these new technologies so that they can keep pace with rivals. From the point of view of a **project manager**, this may result in a client who regularly changes their mind about what they want from an IT project, which, if not managed correctly, can result in missed deadlines and escalating costs.

Managing a project

Managing an IT project can be a difficult and complex job. There are often many stages and a number of different **stakeholders** who have to be involved and managed as part of the process. The project manager's job is to make sure that all stakeholders are involved at the correct time and in the correct way. The project manager must also manage client expectations, ensuring that while the project delivers what they want and need, the client is also aware of what is viable given the time, budget, staff and technologies available. Some of the key features of managing a project are listed here.

Complexity management

IT projects often involve tasks that affect each other, and use a range of technologies and stakeholders. The goal of effective project management is to break down the complexity into manageable tasks, ensuring that each aspect is properly planned, executed and monitored.

Resource allocation

Project resources are anything with a tangible value in the project – including human resources such as software developers, physical resources such as computer hardware or more conceptual resources such as time and budget. Each of these resources has to be carefully considered and will affect the project management. The project manager will need to schedule tasks, and assign responsibilities to specific personnel (human resources), ensuring that they are used effectively, and make sure that the appropriate time, budget and physical resources are available for a task to be completed effectively.

Risk mitigation

IT projects can be very risky because of their complexity and wide range of contributory factors. Project management involves identifying potential risks to the project, developing **mitigation** strategies and implementing contingency plans to minimise their impact on project success. Risks in an IT project can include those listed here.

Changing requirements – clients often wish to add additional features (**feature creep**) or even change the focus of a project while in development. Although some variation is to be expected, a project manager must ensure that the project remains as close to the original project scope as possible. Where a significant change is required or expected, the project manager must ensure that the client and other stakeholders are clear as to what is to happen and the impact this will have on the project as a whole.

Technical issues and threats – these can be as simple as faulty hardware that delays the installation of a piece of software for a day or two, or more serious issues such as a data breach, which could have wider impacts for a company and not just the project itself.

Human factors – a project manager has to oversee and use human resources, including the potential issues that come with this. Issues can include incorrectly configuring hardware or software or mistakenly sharing/deleting sensitive data. Other human issues to consider include allocating the correct person to a job (e.g. ensuring that they have the skill and the time to complete it effectively) as well as dealing with issues such as illness/absence while keeping the project on track and moving forward.

Wider issues – IT projects can also be affected by a range of issues that may be beyond the project manager's control. These can include global issues such as a financial market crisis or a global pandemic, or more localised/regional issues such as changes in local legislation or political instability. Although these factors may be out of a project manager's control, it is important that potential issues are analysed and planned for.

SKILLS

REASONING, COMMUNICATION

ACTIVITY 1

Rupa owns a restaurant located in a busy city centre. She wants to upgrade the computer systems she uses to a system that integrates payments, bookings and stock management.

With a partner, discuss what complexities or issues the project manager might need to consider when first starting the project.

As you discuss the issue, make notes to capture what you have discussed.

Use these questions to help you begin your discussion (but note that you are not limited to these):

- What hardware and software might be needed?
- When will the system be installed?
- What other information do you think you would need to plan this project?

KEY TERMS

project lifecycle: the main phases of a project (e.g. initiation, planning, execution, monitoring, closure)

quality assurance (QA): the process of ensuring that the desired level of performance, design and reliability is maintained

Quality assurance

As well as ensuring a project is completed, effective project management delivers the project in a way that meets the agreed standards and specifications, in order to confirm **quality assurance (QA)**. A project manager must oversee quality assurance at all stages, making sure that effective systems are in place throughout the **project lifecycle**.

Communication and collaboration

IT projects typically involve diverse teams, including developers, designers, testers and stakeholders. The project manager must make sure that communication and collaboration among team members is effective. The project manager must also ensure that the project goals, priorities and expectations are effectively communicated to all stakeholders.

Change management

IT requirements often evolve over time because of changing business needs or market conditions. Effective project management provides mechanisms to assess the impact an IT project will have on the organisation and stakeholders. Where appropriate, the project manager should facilitate mechanisms to prioritise changes and incorporate them into project plans in such a way that disruptions are minimised, and staff and wider stakeholders are supported in the transition.

Scheduling and deadlines

IT is a fast-paced industry, so ensuring that deadlines are met and schedules followed is key to delivering a project in a timely manner. IT projects often consider the concept of 'time to market'; that is to say, the speed at which the product can be delivered and used by the organisation. Ensuring that this is done as rapidly as possible is critical for gaining a competitive edge. Effective project management helps streamline development processes, identify bottlenecks and prioritise tasks so that a project is completed swiftly without compromising quality.

CHARACTERISTICS OF A SUCCESSFUL IT PROJECT

IT projects can be quite diverse in terms of their focus. Sometimes they might be large-scale hardware based projects, while at other times they may involve production of a relatively small-scale software application to solve a very specific problem. Whatever the focus of the project, there are some key characteristics that help to ensure an IT project is successful.

Clear objectives

It is important to clearly define what should be achieved as a result of the project. Clearly defined objectives help all stakeholders to identify what is required and help to inform the planning, execution and evaluation of project tasks. It is common to use SMART objectives (covered in more detail later in this topic) to provide clearly defined and measurable objectives.

Effective stakeholder engagement

Stakeholders are individuals or groups who have an interest in a project's outcome. Their interest can be because of potential positive or negative impacts of the project. Stakeholders will be affected by the project's success or failure and can influence its progress in various ways. Successful IT projects will involve stakeholders throughout the project, either to gain input and feedback or to ensure that the stakeholder is well informed and supported through the change that the project will cause. Stakeholders are typically divided into internal and external stakeholders.

Internal stakeholders: these are stakeholders in the organisation(s) who are directly involved in the project, such as the development company and the organisation that has commissioned the IT project. The stakeholders in these organisations will include development team members, staff of the company that will use the system, managers and executives.

External stakeholders: these are stakeholders who are not directly involved in or employed by the organisation but may have a direct or indirect interest in the project. They may include the organisation's customers/clients, members of the local community, government agencies and regulatory bodies.

SKILLS

ANALYSIS, ADAPTIVE LEARNING

ACTIVITY 2

In Activity 1 you considered this scenario:

Rupa owns a restaurant located in a busy city centre. She wants to upgrade the computer systems she uses to a system that integrates payments, bookings and stock control.

Now make a list of who the stakeholders would be in this scenario. For each stakeholder write a few sentences explaining what their interest in the project would be and how they would be affected (positively/negatively) by this project.

Detailed planning

IT projects can have many components that need to be considered. Therefore, a successful IT project must be carefully and accurately planned. Detailed project planning involves comprehensive project plans that outline scope, deliverables, timelines, resource requirements, risks and quality standards. It is also important that scheduling, budgeting and resource allocation are realistic to ensure project feasibility and success (see the case study on mobile app development later in this section for an example of how project planning can work).

Skilled team

A successful IT project requires a skilled and motivated team with the necessary expertise to effectively execute project tasks that they are allocated. The project manager may select team members who have the required skills and knowledge needed for that project. Where a team is already established, the project manager will allocate project tasks that best match their ability and experience.

Adaptability

Successful IT projects incorporate a degree of flexibility into their plans so that they can adapt to evolving requirements, technologies and market conditions. However, it is important to ensure that while plans accommodate changes where they are needed, the main project objectives are not ignored and that feature creep is limited.

Effective communication

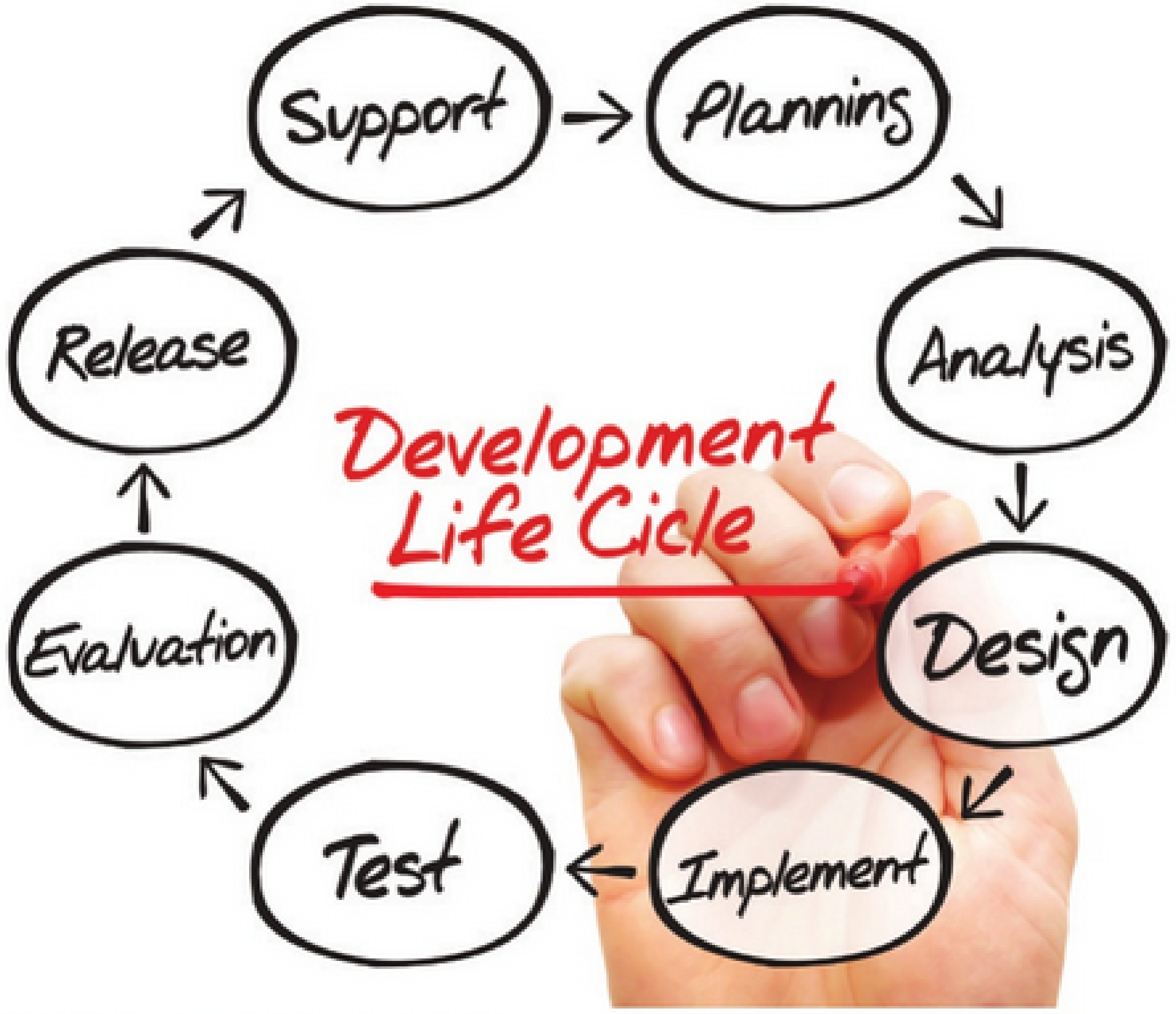
Clear and open communication is essential to ensure a project is successful; the project manager should establish effective communication channels among team members, stakeholders and project leadership. As part of the project planning phase, it is good practice for the project manager to provide clear guidance on when and how members of the team will communicate with each other and key stakeholders. This should include regularly scheduled team meetings, agreed meetings with the client, and guidance on how the development team members communicate progress on tasks. Effective communication ensures alignment between teams, helps to effectively avoid or resolve issues, and manages expectations.

Quality focus

Ensuring that outcomes are of agreed and expected quality throughout the project lifecycle is vitally important to ensure the expectations of stakeholders are met, and potential issues are minimised. An effective project manager will, in consultation with other stakeholders, define quality standards that will govern the project. The project manager must then ensure that appropriate processes and best practices are in place to deliver high-quality solutions that meet the agreed project outcomes.

Risk management

To ensure the success of a project, it is important that the project manager develops appropriate risk management plans. These should outline strategies for identifying, analysing, prioritising and responding to risks to minimise their impact on project success.



▲ Common stages of an IT project lifecycle

Continuous improvement

Successful projects place great emphasis on continuous improvement and learning. Throughout the project it is important to carry out reviews to identify areas of success and areas of improvement to ensure that project objectives are met. At the end of a project, teams should reflect on the project and identify lessons learned to avoid similar issues in the future, and identify things that have been done well in order to apply best practices to future projects.

On-time and on-budget

In addition to meeting the agreed system requirements, two key requirements for the success of a project are that it is completed within the agreed time scale and within the agreed budget. Planning is vitally important to ensure that this happens, and the project manager should use these plans to track progress against milestones, monitor how the budget is being spent and, where necessary, take corrective actions.

THINKING LIKE AN IT SPECIALIST

It is common for many IT professionals to use risk analysis when planning or evaluating IT systems and solutions.

There are different ways to analyse risk, but a numbered risk matrix is the most common. In a numbered (3 × 3 or 5 × 5) risk matrix, the project manager or IT professional will identify potential risks and assign each risk a score for:

- how likely it is to happen (probability)
- the level of severity or consequences if it does happen (impact).

These two scores are then multiplied to give a risk score, which indicates the level of risk. The risk score can be used by managers to decide whether or not to proceed.

Here is an example of a 5 × 5 risk analysis scale: the first part shows the probability and impact, and the second part shows the risk score.

	Probability		Impact
1	Highly unlikely	1	Insignificant
2	Unlikely	2	Minor
3	Possible	3	Significant
4	Likely	4	Major
5	Almost certain	5	Severe

Risk Score	Level of Risk
1–4	Very low
5–9	Adequate
10–16	High
17–25	Severe

Do some additional research on this and other risk analysis methodologies.

Which methodology would you prefer to use to analyse the risk of an IT project? Why do you prefer that methodology?

SKILLS

ANALYSIS, ADAPTIVE LEARNING, PROBLEM SOLVING

ACTIVITY 3

Scenario

A small local village medical centre wants to upgrade its IT systems.

Appointments and patient records are currently managed on a single computer in the medical centre's admin area by admin support staff.

To make an appointment, patients must call the office to speak with a member of the admin support staff.

Each day the admin support staff print out the doctor's appointment list, along with copies of any relevant patient information. The doctor manually records notes from the appointment (e.g. diagnosis, prescriptions, etc.) on a form. At the end of the day, the admin support staff enter the notes from the form into the computer and then shred any physical copies generated for that day.

The senior doctor wants to upgrade the IT systems so that:

- doctors can access and update patient records directly from a computer in the consultation rooms
- the appointment list is distributed digitally to doctors (e.g. on a digital calendar)
- patients can make appointments online.

Task

Use an appropriate risk analysis methodology to:

- identify potential risks of the IT upgrade
- analyse the level of risk for each of the identified risks
- suggest ways that the identified risks could be reduced.

SMART TARGETS

SMART targets are an approach to setting goals and objectives for a project in a way that ensures they can be used effectively to monitor and evaluate the project. The term SMART target is an acronym based on the desirable features of effective target setting. SMART targets must be:

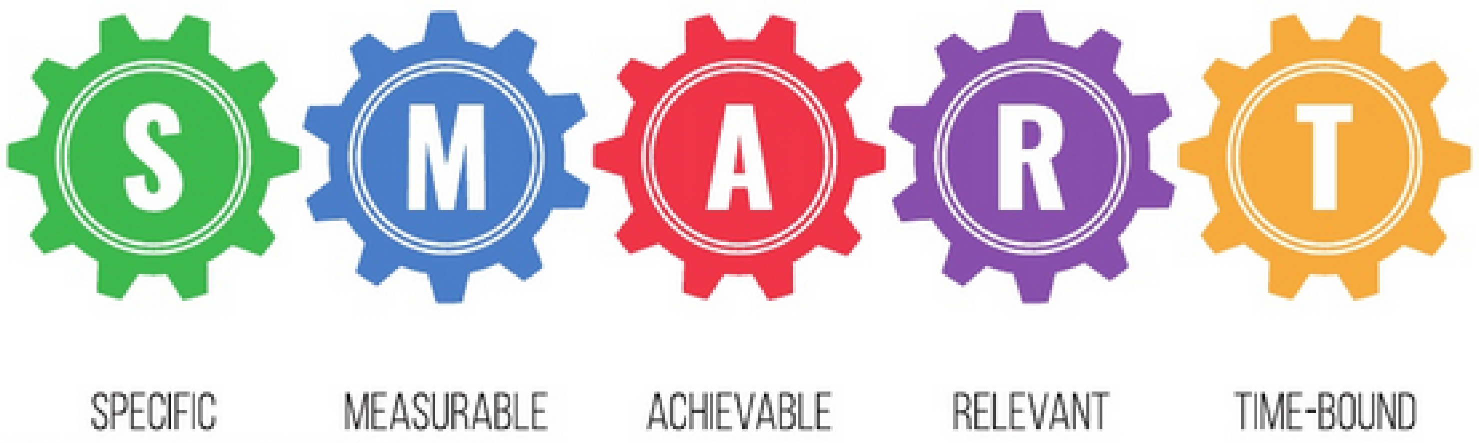
Specific – the goal must be clear and well defined so that it can be more easily tracked and assessed.

Measurable – there must be some way to quantify or measure success/progress of work towards the goal.

Achievable – it is important to ensure that the goal can be achieved. It is often helpful to break tasks down into smaller more manageable sub-tasks. However, a project manager should be careful not to subdivide a target so much that the team loses sight of the overall objectives of the project.

Relevant – each target should align with the overall project aims. If it is not clear how a target links, try rephrasing it or provide additional information to relevant stakeholders to make sure they know the purpose of the target.

Time-bound – every target should have a clear and appropriate deadline. Open-ended tasks result in a lack of urgency or focus. Unrealistic timelines place team members under too much pressure, resulting in low morale and increased errors.



▲ Effective targets follow SMART principles

There are many ways to present project objectives and SMART targets, and each company and project manager will have their preferred way to do this. The important thing to remember is that the targets are presented in a way that ensures they meet SMART principles, and that all relevant information is easy to understand.

SMART example 1

Objectives can be listed along with a space where they can be reviewed/assessed at a later date (see Table 15.1). This method is useful for personal objectives, such as for a staff appraisal or list of tasks for a project. It is also useful for less complex projects that have fewer objectives and require fewer sub-tasks to be completed. In more complex projects with a large number of objectives, presenting them in this way can make it difficult to see the links between them; it is also more difficult to plan the specific resources required to achieve the objective.

TABLE 15.1: PRESENTING PROJECT OBJECTIVES (BASIC)

Objective	Completed (Yes/No)	Review comments
Complete training course in application of security protocols by next monthly review meeting.		
Complete testing of user account module and integrate with website. (Deadline – 25th March)		

SMART example 2

In a project with multiple team members and a series of objectives that are interrelated, it is useful to present the objectives in a way that allows the project manager to see the whole picture (Table 15.2). The project manager should identify who is responsible for delivering the objective, what other resources are required and how they link to other tasks and objectives.

TABLE 15.2: PRESENTING PROJECT OBJECTIVES (WITH MORE INFORMATION)

Objective	By whom?	Deadline	Resources required	Additional information
Install network cabling in main office	Senior Network Engineer	13 December 2024	100m Cat6 cabling Network access points	Requires new server and switch to be in place.
Install new server in server room	Senior Network Engineer	13 December 2024	New file server Network switch	Old server must already be removed and backed up. Backed up data to be transferred to new server.
Install and configure admin staff PCs	Junior Network Engineer	21 December 2024	6 × standard office PC	Cabling, switch and server upgrades must be complete. PC software must be purchased, delivered and be ready to install.

SKILLS

SELF-MONITORING, SELF-DIRECTION

ACTIVITY 4

Think about a substantial piece of work or task you must compete in the coming weeks.

What does the task involve? What must be achieved to ensure that it is completed to the best possible standard?

Break the work/task down into a series of smaller tasks that will need to be completed.

Write a set of SMART objectives that will allow you to track your progress through the work and assess the quality of your final outcome.

PROJECT MANAGEMENT TOOLS

A successful project has to be carefully planned and monitored throughout its whole lifecycle. This includes timings, task progression, budget and resource usage. A project manager will use a range of tools as part of the project management process. These tools are often packaged in specialist project management software, but whichever software package is used, the basic principles of the tool itself are the same.

KEY TERM

network diagram: a diagram that provides visual representation of project tasks, dependencies and prerequisites (e.g. arrow diagram method, Gantt chart)

Nodes and Gantt charts

Nodes and Gantt charts are fundamental tools in project management; although the context in which they are used is slightly different, they are related and will often be used together to support project planning and tracking.

A node refers to a point or milestone in a project schedule or **network diagram**. It is used to represent significant events, tasks or deliverables that mark progress or completion stages in the project. Each node relates to a **specific** and **measurable** target or objective.

A Gantt chart is a visual representation of a project schedule that shows start and end dates of individual project tasks. The Gantt chart clearly shows how different tasks relate to each other by showing the order that tasks are to be completed, and linking activities that depend on previous tasks. This is also useful for identifying tasks that can be completed simultaneously. In a Gantt chart, tasks are represented as horizontal bars along a timeline, with each bar indicating the duration of the task. Bars are positioned relative to their start and end dates, providing a visual representation of the project timeline. Gantt charts often include additional information such as task dependencies, milestones, resource assignments and progress tracking indicators.

THINKING LIKE AN IT SPECIALIST

Gantt charts can be created using standard office productivity software such as a spreadsheet. However, for more complex projects it can be difficult to keep track of allocated resources, such as team members.

Project management software typically provides Gantt chart functionality with additional features such as alerting the user when there is a conflict in resource allocation.

Project management software can be quite expensive, but open-source alternatives such as ProjectLibre (community edition) can provide more affordable options.

Do some research into open-source project management software. Download the software and explore the tools provided.

Requirements

Defining project requirements is an important step in the project management process that involves identifying, documenting and communicating the needs of stakeholders and the project's deliverables and outcomes. The project requirements are closely linked to the SMART objectives covered earlier in this section: the requirements define the overarching, high-level objectives of the project and typically define the final product to be delivered. The requirements will typically be further broken down into smaller tasks that will form the project's SMART objectives.

When defining the requirements of an IT system it is usual to present these as functional and non-functional requirements.

Functional requirements

These requirements define exactly what the user(s) should be able to do in the IT system. It means providing details of the specific functions and features that will be available, and provides the scope for the deliverables of the project. For example:

User authentication:

- the system must allow users to log in with a username and password
- after the username and password are provided the system requires additional authentication from the authenticator app.

Inventory management: the system must track the quantity and location of inventory items.

Reporting: the system must generate reports detailing user interaction rates for management.

Non-functional requirements

Non-functional requirements specify how the system or product should perform. They allow the project manager to define the quality attributes, constraints and characteristics of the system. They are used to define deliverable aspects of the project such as performance, security, usability, reliability and scalability. For example:

Performance: the system must respond to user input within two seconds.

Security: the system must use public key encryption to protect transactions.

Usability: the system interface must be intuitive and accessible to users with disabilities.

Reliability: the system should have an availability of 99.9 per cent during business hours.

Scalability: the system must support spikes in the number of concurrent users without loss in performance.

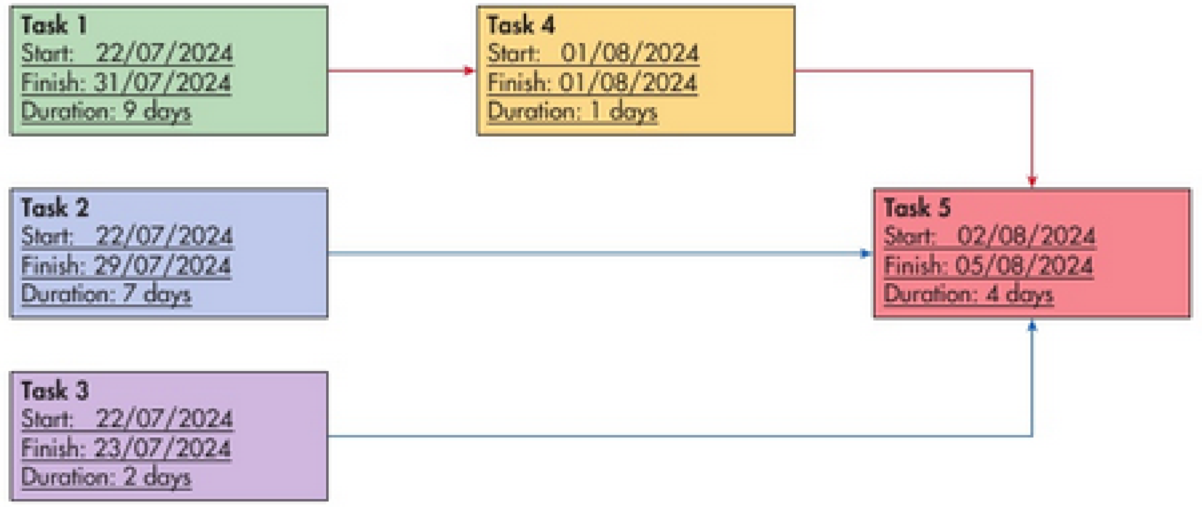
Critical path analysis

Critical path analysis is used to identify the sequence of tasks that must be completed on time to ensure that the entire project meets the specified deadline. It allows project managers to identify those tasks that are critical to the project and cannot be delayed. This enables the project manager to prioritise the most important tasks and allocate time and resources to those first. Other tasks that are not on the critical path can be allocated some slack (also known as buffer time), which can allow them to be delayed or reorganised to ensure that the tasks on the critical path are completed on schedule.

To perform a critical path analysis, the project manager must:

1. Identify *all* the individual tasks needed to complete the project.
2. Estimate the time it will take to complete each task (duration).
3. Identify any dependencies between tasks (e.g. back-end database must be completed before integration testing for the e-store can begin).
4. Present the tasks using a suitable diagram (e.g. Gantt chart, arrow diagram method).
5. Work out the longest sequence of tasks that are dependent on each other – this is the critical path.

Figure 15.1 shows a simple network diagram for a project, with the critical path shown by the red arrows. Even a one-day delay to the tasks in this path will affect final delivery date of the project. A one-day delay to Task 2 or Task 3 will not affect the overall completion of the project.



▶ Figure 15.1 A network diagram showing the critical path

Once the critical path analysis is complete, the project manager should refer to it and refine it continually over the course of the project.

1. The project manager must allocate slack to any tasks not on the critical path. These tasks can be delayed or rescheduled with minimal impact on the final deadline.
2. The project manager must monitor the progress of all tasks, but particularly any tasks identified as being on the critical path.
3. If delays occur, then any impact on the project timeline should be considered and adjustments made to the project plan. This may mean reallocating resources from less critical tasks to those on the critical path until the project is back on schedule.
4. Where serious or critical issues occur, the project manager will need to alter deadlines for non-critical tasks in the first instance; if this does not solve the problem, it may be necessary to negotiate adjusted timelines with the client, or in extreme cases even change the project scope.

SKILLS

PROBLEM SOLVING, DECISION-MAKING

CASE STUDY: MOBILE APP DEVELOPMENT

A project manager for an app development company is analysing the critical path for developing and launching a new mobile app. The table here shows the main tasks required along with their durations and dependencies.

Task ID	Task Name	Duration (days)	Dependencies
1	Planning and requirements gathering	5	[]
2	UI/UX design	10	[1]
3	API development	12	[1]
4	Mobile development (iOS)	15	[2]
5	Mobile development (Android)	14	[2]
6	Testing and quality assurance	8	[3, 4, 5]
7	App Store deployment	3	[6]
8	Play Store deployment	3	[6]
9	Marketing and promotion	4	[7, 8]

The project manager has identified the critical path for this project as:

- Planning and requirements gathering
- UI/UX design
- API development
- Mobile development (iOS)
- Mobile development (Android)
- Testing and quality assurance
- App Store deployment
- Play Store deployment
- Marketing and promotion.

The total duration of the critical path is 74 days.

Making decisions

The project has incurred a massive delay during the planning and requirements gathering and UI/UX development stages of the project.

With a partner, discuss what changes could be made to the project scope and objectives, so that the project manager can deliver a working app and reduce the amount that they go over the original deadline.

Precedence tables

A precedence table is a visual tool that is helpful for scheduling and sequencing tasks (see Table 15.3). It shows the order in which tasks must be completed, including any dependencies between them. This information is crucial for understanding the project's overall timeline and potential bottlenecks. The precedence table gives the duration of each task and when it must be started and finished. This enables the project manager to identify critical tasks that affect the overall timeline, which can improve resource allocation as priority can be given to more crucial tasks.

Precedence tables can become quite complex, especially for large and intricate projects with numerous tasks and dependencies. Their effectiveness is also dependent on accurate estimates of task durations and dependencies, which can be challenging for less experienced project managers. It is therefore important that a project manager has either good knowledge of the project domain, or that they have communicated effectively with the senior developers and engineers in the team to get their input on task duration.

Key features of a precedence table are as follows:

Tasks: each row represents a specific task in the project.

Precedence: indicates which tasks must be completed before others can start (dependencies). For example, in Table 15.3, Task 4 (Wireframing and prototyping) has precedence 2, 3. This means that Task 2 (Content audit and revision) and Task 3 (Information architecture development) would have to be completed before it can start.

Duration: estimates the time needed to complete each task. This is usually presented in days. For each project the length of the day (in hours) would be decided by the project manager and would be influenced by the teams' standard contracted hours.

Early start (ES): the earliest possible start date for a task, considering its dependencies. Table 15.4 shows the number of days into the project that this will happen. For example, Task 2's earliest start will be 5 days into the project because it is dependent on Task 1, which will take 5 days to complete.

Early finish (EF): the earliest possible finish date for a task. This is calculated by adding the duration to the ES.

Late start (LS): the latest possible start date for a task without delaying the project completion. This value is equal to the LS of the preceding task. If there is more than one preceding task, the lowest value should be used.

Late finish (LF): the latest possible finish date for a task, calculated by adding the duration to the LS.

Slack: the difference between the LS and ES. Calculated by subtracting ES from LS. This identifies potential flexibility in the task's schedule.

TABLE 15.3: EXAMPLE PRECEDENCE TABLE

Task	Precedence	Duration (days)	Early Start (ES)	Early Finish (EF)	Late Start (LS)	Late Finish (LF)	Slack
1. Gather requirements and planning	–	5	0	5	0	5	0
2. Content audit and revision	1	7	5	12	9	16	4
3. Information architecture development	1	8	5	13	8	16	3
4. Wireframing and prototyping	2, 3	10	13	23	19	29	6
5. Visual design and branding	2, 3	12	13	25	21	33	8
6. Front-end development	4, 5	15	23	38	32	47	9
7. Back-end development and integration	4, 5	18	23	41	34	52	11
8. Content migration and integration	2	10	25	35	42	52	17
9. Testing and quality assurance	6, 7, 8	8	38	46	47	55	9
10. Launch and deployment	9	3	46	49	55	58	9

SKILLS

PRODUCTIVITY

ACTIVITY 5

A company is upgrading its IT systems. Following the initial meeting, a list of tasks and estimated durations has been produced for the project, as shown in the table.

Task number	Description	Duration (days)	Resources required	Notes
1	Upgrade network infrastructure	6	Ethernet cabling, network access points, network switch	Can be completed by the Junior or Senior Network Engineer
2	Install and configure network server	5	Network server	Can only be completed by the Senior Network Engineer
3	Install and configure user workstations (hardware)	3	3 × workstations Power outlets	Server and network infrastructure must be completed Can be completed by the Junior or Senior Network Engineer
4	Populate (back-end) product database from archive	4	Database Data archive Secure development environment	Completed by the Senior Software Engineer
5	Migrate database to server	2	Database Secure remote connection	Server and infrastructure must be fully installed and working

(Continued)

Task number	Description	Duration (days)	Resources required	Notes
6	Install client software to workstations	1		Workstation hardware must be installed and configured Completed by the Senior Software Engineer
7	Testing of system hardware	3		Completed by the Senior Network Engineer
8	Testing system software	3		Completed by the Senior Software Engineer
9	System handover and training	2	System documentation	System must be fully working and signed off by both senior engineers Senior Software Engineer to provide training on using the system

Use your understanding of project planning and IT systems to produce a set of project planning documents for this task.

You should produce:

- a risk analysis
- a Gantt chart
- a critical path analysis
- a precedence table.

Note: Your risk analysis should use any appropriate risk matrix, and should include information about how identified risks can be reduced.

WATERFALL PROJECT METHODOLOGY

The waterfall project methodology is a traditional way of organising the main stages of a project. It is a sequential, linear approach to the development of IT systems that divides the project into distinct phases. Each phase of development is dependent on the outcomes of the previous phase. The key characteristic of the waterfall methodology is that each phase must be completed before moving on to the next one. Because of this there is little room for change in the project scope, as doing so can mean having to start the project from the beginning. However, the waterfall approach is very useful for small or simple projects with few requirements, and also for projects where the requirements are clearly defined and unlikely to change. For example, a project that requires the upgrading of a server and related network infrastructure in an office could follow the waterfall method as it is unlikely that there would need to be much variation in the requirements of the project. However, the development of a new and complex app for a large bank is highly unlikely to follow the waterfall method.

PHASES OF THE WATERFALL METHODOLOGY

The waterfall methodology is made up of distinct phases that separate the project into well-defined sections. Each of these phases is self-contained, and all tasks must be completed within that phase before moving on to the next phase. The phases for the waterfall methodology are as shown in Figure 15.1.

Requirements gathering/analysis: this phase involves analysing the problem and communicating with the key project stakeholders to establish the requirements for the project. It includes identifying what the system needs to do and any constraints it must operate within.

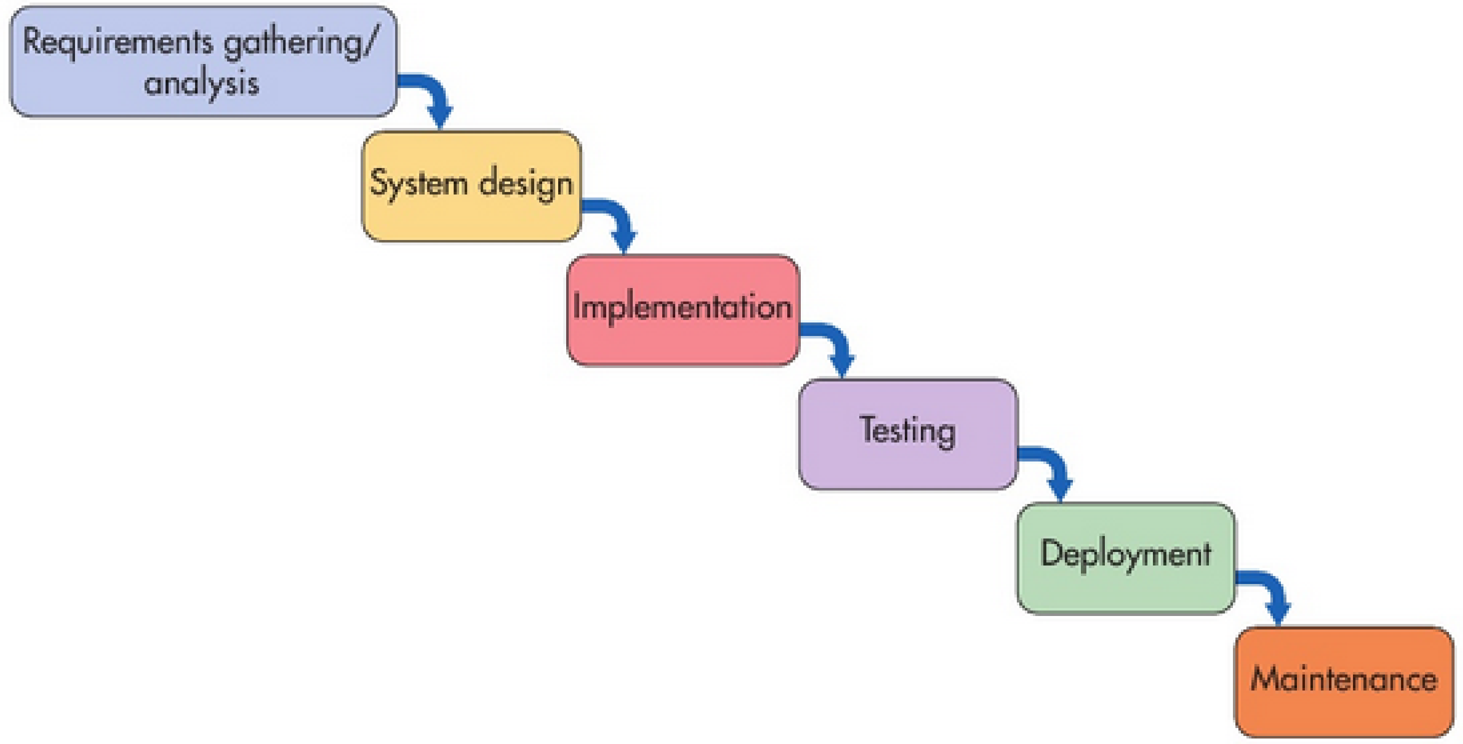
System design: after requirements are gathered, the system design phase begins. Planning expands to develop detailed designs of all aspects of the system, including visual and technical components.

Implementation: sometimes also referred to as the development stage, this stage is where the project team use the designs created in the previous stage to create the system. This stage can include coding software, configuring and installing applications or upgrading infrastructure.

Testing: when the implementation phase is done, the system must be tested. The approach to testing will vary depending on the type of system. The goal of this stage is to ensure that the system is robust, reliable and meets the project specifications.

Deployment: after successful testing, the system is released to the end users and used in the real-life business context for which it was designed.

Maintenance: once the system is deployed, an IT development team will continue to monitor it for an agreed period of time. This involves fixing any bugs that arise, making updates or enhancements as needed, and providing ongoing support to users.



▶ Figure 15.1 Waterfall methodology

WATERFALL METHODOLOGY – TASKS AND ACTIVITIES

During each phase of the waterfall methodology, a project team will perform a variety of activities. Some of the activities carried out may vary from project to project depending on the purpose and focus of the project, particularly with testing, but many of the activities are standard and would be conducted for every project.

Requirements gathering/analysis phase activities

As you saw in the project management section, the key to success of any project is to have clearly established requirements. While very similar, the requirements gathering in this methodology typically applies only to the technical phase, that is, the development of the system itself and not the project as a whole.

Identify stakeholders

It is essential that the key stakeholders have input into what the system must deliver. This will typically be management, who will know what they want the system to achieve for the business as a whole, and potential users who would be consulted on specific functionality.

Gather requirements

Requirements can be gathered in different ways. Table 15.4 shoes an overview of some of the key methods for gathering information from stakeholders, and some of the benefits and drawbacks of using these methods.

TABLE 15.4: METHODS FOR REQUIREMENTS GATHERING

Method	Description	Benefits and drawbacks	
Interview	Direct structured conversation with stakeholders. Typically held in a one-to-one setting	+	• Direct interaction with stakeholders • Can pick up on nonverbal cues as well as specific answers • Can use follow-up questions to gain clarification to answers • Can help build good relationships with stakeholders.
		–	• Can be very time consuming due to one-to-one nature and issues with scheduling • Some stakeholders may feel uncomfortable in this type of setting • Difficult to ensure that collected information is structured in a way that can be analysed easily and effectively.
Observation	Directly watching stakeholders in the intended environment. For example, watching an employee use the current system	+	• Able to see stakeholders in a real-life environment • Can more easily identify current processes, challenges and user needs.
		–	• Can be obtrusive and disruptive • Can make the stakeholders feel uncomfortable • Can lead to assumptions and generalisations being made by the observer.
Document analysis	Reviewing existing documentation such as business reports, user manuals or regulatory documents to extract requirements	+	• Provides business specific data and information that provides valuable context • Highlights key requirements and challenges such as industry specific guidelines that must be followed.
		–	• Quality of documentation can vary • Documentation may not capture actual established practice • Industry-specific guidelines and technical aspects may be misunderstood by non-specialists.
Focus group	Group discussions and workshops involving stakeholders in a collaborative setting	+	• More open nature of the format can lead to more natural idea generation • Stakeholder-led activities are more likely to focus on key requirements • Helps build good relationships with stakeholders.
		–	• Can be difficult to capture discussions in a way that makes them easy to analyse • Discussions can be highjacked by particular personalities, or by individuals with specific agendas, leading to skewed results.
Questionnaire	Structured prewritten questions distributed in paper-based or online forms	+	• Very structured – format of responses will be consistent from all participants making it easier to collate and analyse • Can be distributed to a wide range of participants • Participants can complete it at a time and speed that suits them • Cost effective way to illicit feedback from large groups of stakeholders.
		–	• Very structured – format of responses will be consistent from all participants making it easier to collate and analyse • Can be prone to low response rates • Participants may misinterpret a question and cannot seek clarification • Can lack depth of other forms of information gathering such as interview.

(*Continued*)

TABLE 15.4: METHODS FOR REQUIREMENTS GATHERING (*Continued*)

Collate and analyse gathered requirements

The information gathered from key stakeholders should be carefully analysed and reviewed. The analysis should look for common themes and trends so that data-driven decisions about what is needed can be reached. It is important to also identify any conflicts or gaps in the requirements and check why this is. It should not be assumed that because a requirement has not been raised at this early stage, it is not important. It may be that the method used to gather the information was flawed, or that incorrect stakeholders were targeted, or that not all data has been collected.

Ensuring quality of gathered data

It is important that the data gathered is of high quality so that it leads to appropriate conclusions. When gathering data to inform the project requirements there are some key principles that should be followed to ensure the quality of data.

Select the right stakeholders. It is important that a broad range of stakeholders are represented. Over-representing or under-representing particular groups can skew results and provide an incorrect focus for the project; for example, only involving senior managers that see end reports and results, but who will never use many of the functions of the system, could lead to a system that is difficult to use for everyday users.

Use a mixture of closed and open questions. Closed questions can provide quantitative data that can be more easily analysed. Open questions can provide detailed responses that lead to a deeper understanding of the problem.

Provide structured questions for interviews so you can make comparisons between interviews. Provide guidance and proforma that can be used to record additional or follow-up questions and discussions that occur so they can be captured in a way that can be analysed consistently.

Select the most appropriate collection method for the targeted group. For example, a long and complicated questionnaire would not be appropriate to get the opinions of a learning platform designed to be used by young children.

Gather team

Once the requirements have been established, the project manager or IT lead will identify the team members who will be used for the project. Most IT development companies have team members who specialise in particular areas. For example, if a project is heavily reliant on infrastructure upgrades, then the team will contain more hardware engineers, or the project may require team members who have web development skills and experience deploying systems in the cloud.

System design phase activities

The system design phase is where the look, content and functionality of the system is planned. This section will cover some of the key design documents and activities that are used in an IT project. The types of design and documentation required at this stage will vary depending on the type of system being developed. For example, an IT project that requires only installation of hardware and the use of personal computers, with preinstalled operating systems and productivity software, would use different documentation from a project involving the design of a new customer-facing website and e-commerce store.

User experience design

User experience (UX) design should be considered alongside user interface (UI) design but it is important to know that they aim to achieve slightly different things. UI design focuses on the look and style of the system and the way in which the humans and computers interact with each other. UX design is an area of the design phase that focuses on ensuring products and systems provide meaningful and enjoyable experiences for users. The goal of UX design is to ensure that the IT system is designed so that the look and functionality of the systems is supported by usability and accessibility considerations for user satisfaction and continued use. In addition to the research carried out in the requirements gathering stage, UX design makes use of some key documents and tasks.

User personas: these are fictional representations of typical users based on research data. They are used to provide designers with reference points for the main users of the system, their preferences and what they will expect from the system in different scenarios.

Information architecture (IA): this involves organising and structuring content and functionality in a way that aids intuitive navigation, as well as efficient and effective access to required information. This often includes designing site maps, navigation menus and content hierarchies to plan how data will be presented and related to each other.

▶ Ensuring the interface and information work together is key to successful UX design

Wireframing and prototyping

Wireframes and prototypes are low-fidelity versions of a user interface or product that bridge the gap between UX designs and full UI designs. These designs illustrate the layout, structure and functionality of a product or service, but without the specific colours, fonts, images and so on that will be found in the final product. These designs allow designers to explore different design solutions, gather feedback early in the design process and refine interactions before final implementation.

Visual design

Visual design focuses on the aesthetics and presentation of the user interface, including colours, typography, imagery and final layout (which would be influenced by the findings during the UX design stage). It aims to create visually appealing and cohesive designs that communicate brand identity, evoke emotional responses and enhance usability.

Accessibility

Accessibility ensures that products and services can be used by people of all abilities, including those with additional needs. This involves designing inclusive or alternative interfaces, providing alternative means of access and adhering to accessibility standards and guidelines.

THINKING LIKE AN IT SPECIALIST

The World Wide Web Consortium (W3C) along with the Web Accessibility Initiative (WAI) are organisations that provide guidelines on how to ensure accessibility in web-based products, known as the Web Content Accessibility Guidelines (WCAG).

These accessibility guidelines provide a comprehensive framework for making web content and apps accessible to people with disabilities and are now the 'go to' standards for accessibility for most computer systems.

Visit the W3C website and read more about the guidelines.

Network and system diagrams

Network and system diagrams are used to plan the layout and structure of the physical aspects of the system. They are used to plan where computer systems will be placed, how cabling will be installed and even the number of access and power outlets needed.

Flowcharts

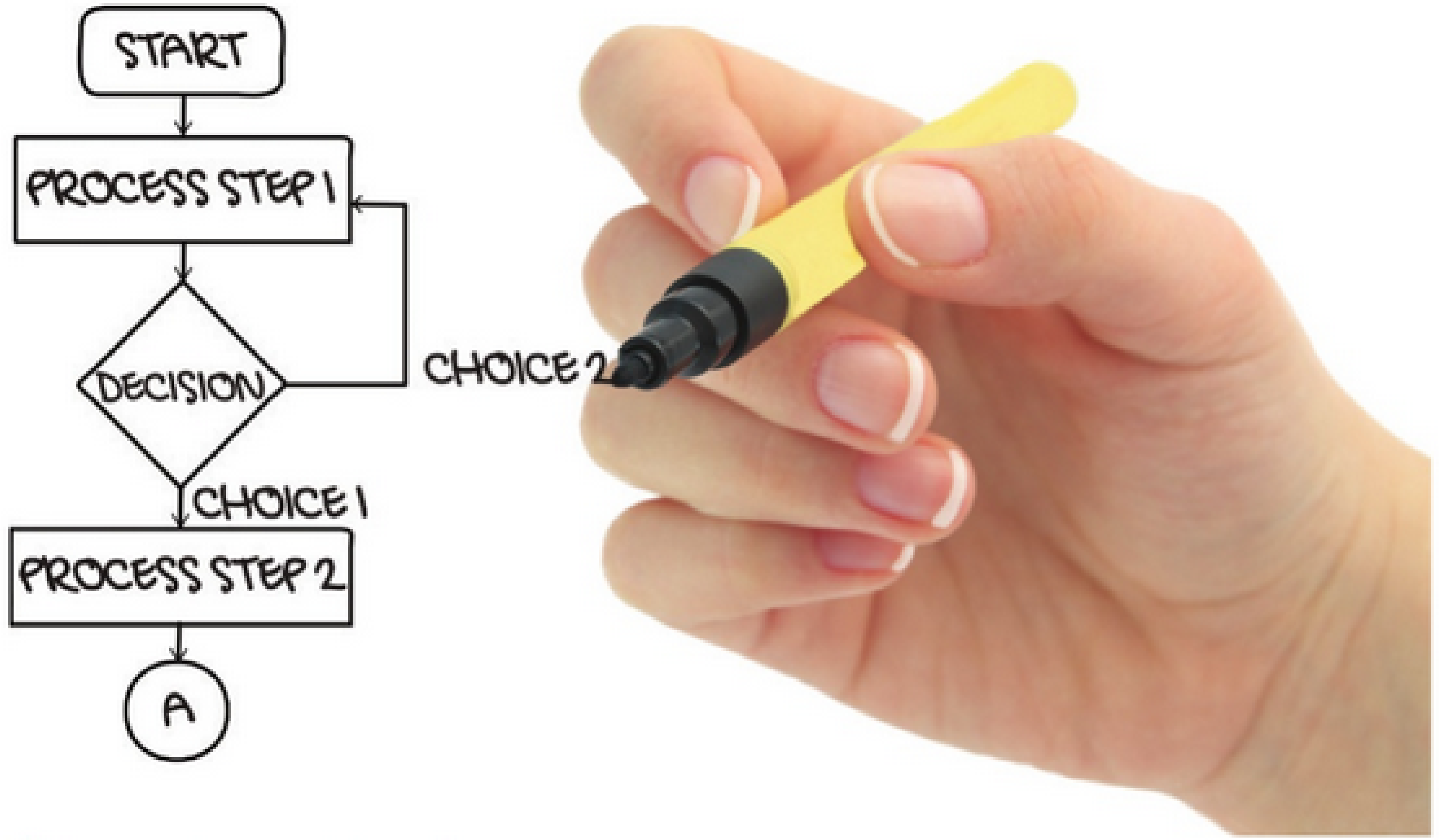
Flowcharts are used to plan the data flow and logic of a computer system. They can be used to represent an overview of a system or one specific part or algorithm. Flow charts are very versatile and visual, and they can be used in different ways. In a system development methodology they will typically be used for the following activities:

Documenting existing processes: flowcharts can be used to capture the steps involved in a process, making it easier to remember, analyse and share with others. This allows developers to record the requirements of a current business process and more easily translate that into functionality for the new system.

Communicating complex processes: flowcharts break down complex processes into simple steps and connect them in a visual manner. This can make the processes involved in intricate systems more accessible to audiences with varying levels of technical expertise. This is useful when explaining the plans for new systems to key stakeholders, such as senior management.

Planning and designing algorithms: flowcharts are often used in software development and computer science to represent the logic and decision-making involved in algorithms visually.

UX and UI design: flowcharts can map out the user journey through a website, application or system, allowing designers to identify the number of steps in a process and the user interaction required, so that bottlenecks and pain points can be avoided.



▲ Flowcharts help to plan the logic of a system or process

Information flow diagrams

An information flow diagram (IFD) is a visual representation that shows how information moves within and between systems or organisations. It helps to understand the sources, destinations and pathways of information, as well as any transformations or processing that occurs along the way.

IFDs are covered in more detail later in this topic.

Implementation phase activities

The implementation phase involves the actual creation of the IT system based on the designs produced in the previous phase. Depending on the IT system being developed, it may involve some or all of the following activities:

Coding: software engineers/developers translate the detailed design documents into working code, following predefined specifications and standards.

Building computer systems and infrastructure: hardware engineers/technicians will combine components, devices and peripherals to form the infrastructure that the system will run on.

Database creation: software/database engineers will build, populate and link databases to other parts of the system to provide back-end system processes for the computer system in order to meet specific business goals and project requirements.

Interface creation: methods of interaction for users will be created and implemented, including accessibility options for those with additional needs, and alternative interfaces to provide responsive layouts for users using different devices (e.g. desktop and mobile versions of a website).

Documentation: the development team will produce supporting documentation for the system. It may include user guides to provide help and support on the functionality of the system, as well as technical documentation with information and data to support future maintenance and improvement of the system.

Testing/debugging phase activities

Once the system is complete, the project will move to the testing phase. During this phase the development team will carry out a series of activities designed to discover any issues with the system that were not detected during implementation. In addition to ensuring the system functions as expected, testing must ensure that all functional and non-functional requirements have been met and that the system is robust, consistent and reliable. The main types of testing used include the following:

Unit testing: individual units or modules of code are tested by developers to ensure they perform their specific functions correctly. Developers will often use automated testing tools to rapidly test the unit/module against different scenarios and different inputs and data.

Integration testing: when more than one unit is completed, the units are combined and tested together to verify that they work together. Integration testing will consider these questions:

- Does the system continue to function as new units are added?
- Do units that are supposed to communicate and share data do so?
- Is the data that is passed between units correct and reliable?

System testing: the entire system is tested against the predefined requirements and user needs. This typically involves manual testing by a dedicated QA team to simulate real-world usage scenarios.

Acceptance testing: this final stage involves formal confirmation by stakeholders or end users that the system meets their expectations and is ready for deployment.

During the testing phase, all tests and the outcomes of these should be recorded, including detailing any corrective actions that have taken place. A detailed log of testing is useful for auditing and quality assurance purposes, as well as for root cause analysis should a bug or issue arise in the future.

Installation phase

System development will usually take place in isolated systems owned and operated by the development company. Once testing has been completed, the system must then be migrated and deployed to the target devices, systems and locations as required, at which point it can be handed over to the client.

The specific activities differ depending on the project. However, some of the key tasks that are likely to take place are listed here.

Installation of physical infrastructure

If the system is going to be reliant on any local, on-premises hardware, then this will be installed first. It is common for the hardware team to prepare this while any software components are in the final stages of testing to ensure that the project is not delayed.

Set up and configuration of cloud-based infrastructure

Once the physical hardware is in place it can be connected to and integrated with any cloud services that are needed. Because of the nature of the cloud services, many of these services will have already been set up and configured during the implementation phase. The developers will just need to make sure that the new hardware is configured so that they can connect to these third-party services without issue.

Installation and deployment of software components

Any new software systems that have been developed will now be installed onto the target machines ready for use by the company.

End-user training

Before a system goes live, the development company will provide key stakeholders with training on how to use the system. Stakeholders will probably be divided into groups and trained only on the area of the system with which they will directly interact. Any in-house IT support staff will be provided with training on a much wider range of aspects of the system, including functionality and basic troubleshooting.

Maintenance phase activities

During the maintenance phase, the development team will provide ongoing support for the organisation on an already deployed product. In the early stages of deployment, maintenance will focus on identifying any issues or bugs that were not identified during testing and installation. As a product becomes more established, the monitoring will become less frequent and focus will be on evaluating whether the product still meets user and business needs in changing situations and environments. Some of the key activities during maintenance include the following:

Bug fixing: when a system goes live it can encounter situations that could not be replicated or anticipated in the testing environment. A development team will typically have a process to allow users to report any issues and defects. These will be analysed and prioritised based on their importance and severity in relation to business goals. It is also common for a development company to have an agreement or contract with the business that defines the process for responding to issues.

Performance optimisation: maintenance is not just about fixing issues but also about improving already working systems to provide more streamlined processes and more efficient and productive users (see the section on perfective maintenance in Topic 14 for more information on this).

Security updates: as security threats are constantly evolving it is important that systems are constantly updated to protect against new and emerging threats. Applying security patches and updates helps to address vulnerabilities as they are discovered.

New feature implementation: the needs of a company change as time moves on, either through changes in focus or through changes in the business landscape. Developing completely new systems can be time consuming and expensive, so where possible it is preferable to add additional features to existing systems. This also reduces the training burden as users will already be familiar with how the rest of the system works.

Documentation updates: it is important to keep user guides and technical documentation up-to-date to reflect changes and new features.

AGILE PROJECT METHODOLOGY

The agile project methodology is a modern approach to project management and IT system development that focuses on small but continuous and rapid improvements to a system. In comparison to the waterfall methodology, agile projects are more suited to large and complex systems, preferring instead to provide systems to end users with fewer features, but much sooner than when following the waterfall methodology.

Many of the individual activities, such as UX/UI design, coding, testing and debugging are the same as those used in the waterfall methodology. The key difference in the two approaches lies in how these are carried out during the project.

KEY TERMS

incremental: small but measurable developments and changes that move a project closer to its completion

iterative: an approach to project/system development where outcomes are continually reviewed and improved

KEY FEATURES AND CONCEPTS OF AN AGILE APPROACH

An agile approach is based on two key ideas, that the project will be **iterative** and **incremental**. The project will continually review and improve the features, functions and quality of the project (iterative). It will also improve and develop continually even if only by a small fraction (incremental).

This approach to iterative development provides a number of benefits, but also presents a number of unique challenges for IT development teams.

Benefits of using an agile approach

Faster time to market: because the number of features in a system is reduced, it is possible to deliver working systems much sooner. This allows features to be tried in a real-world environment much sooner in order to receive feedback and make adjustments.

Increased flexibility: changes that occur are smaller and more frequent, which allows companies to adapt to changing requirements and market conditions much more rapidly as, unlike the waterfall method, the whole project does not have to restart from the beginning.

Improved product quality: continuous feedback, testing and refinement lead to a higher-quality product. Also, because each iteration adds only a limited number of new features or changes, the scope of potential issues is narrowed and so it is easier to identify and rectify problems.

Increased team morale: agile frameworks encourage teamwork and rely on collaboration to be effective, which can lead to more engaged and motivated teams.

Challenges of using an agile approach

Requires cultural change: transitioning from traditional, plan-driven, linear approaches to an agile approach can be challenging, as it can feel less structured and more uncertain.

Uncertain timelines: the focus of each iteration will be determined by the feedback received after it goes live; as a result, managers and developers can be unsure if the next iteration will focus on new features or bug-fixes, so estimating project completion can be difficult.

Requires strong communication and collaboration: success relies on open communication and collaboration within teams and with stakeholders.

THINKING LIKE AN IT SPECIALIST

The agile approach is essentially a set of principles for development; these principles have been used and codified in a number of different frameworks.

The three most common agile frameworks are:

- Scrum: a popular framework with short sprints, daily stand-up meetings and sprint reviews and retrospectives.
- Kanban: focuses on continuous flow of work and visualises tasks on a board.
- Lean: adapts principles from lean manufacturing to IT system development.

Investigate these and other agile frameworks and how they are used in IT system development.

PHASES OF AN AGILE APPROACH

The general phases and activities in an agile approach are similar to those in a waterfall approach. The key difference in an agile approach is that the scope will be much narrower, focusing on small feature or quality improvements. Each phase will also be revisited with each iteration, unlike the waterfall approach, which does not revisit a previous phase once that phase has been signed off.

The main phases of each iteration in agile development can be seen in Figure 15.2 and are as follows:

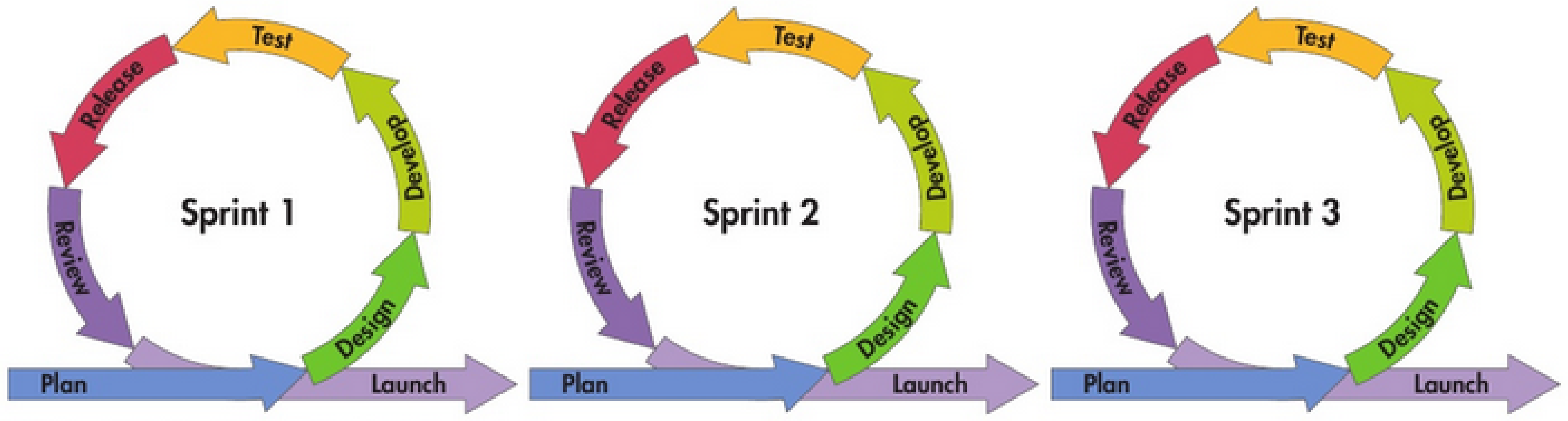
Requirements: at the beginning of a project, as with waterfall, the overall requirements for the project are defined. Then, at the beginning of each iteration, the requirements are revisited, revised as needed and the specific requirements for that iteration are identified.

Plan: an overarching plan is devised at the start of the project, but this is typically less detailed than that of other methodologies. At the beginning of each new iteration, the plans are reviewed and specific plans for that cycle are produced. They will typically include details of the features and functionalities that the team commit to delivering in that iteration.

Design: design phases in agile projects use 'user-centred' design principles, which focus on prioritising the needs and expectations of the user at every stage of the development process. Because agile is incremental and iterative, a just-in-time design approach is applied, meaning that designs are only created as needed and only for the next iteration.

Develop: each development phase focuses on making small incremental changes that have been identified in the requirements, planning and design phase of this iteration. Because the planned changes will typically be relatively small, updates and improvements can be delivered quickly.

Release: at the end of each development phase, the new iteration is released to users. The new iteration is monitored and reviewed for an agreed period and the cycle is repeated.



▲ Figure 15.2 The stages of an agile approach

KEY ACTIVITIES IN AN AGILE APPROACH

Scrum

The term 'scrum' can refer to an overarching framework that provides structures and processes to enable development teams to deliver products in an agile manner. However, it can also refer to a set of key activities. A scrum typically covers the following four key events/activities that inform the development.

Sprint planning: defines the goals and tasks for the upcoming sprint.

Daily scrum: agile methods encourage collaboration and communication between team members, so there is a short daily meeting for the team to share progress and identify potential issues. This helps foster an open environment where members of the team can work collaboratively to solve problems and enable progress.

Sprint review: at the end of a completed increment, feedback is collected from stakeholders and the sprint is flagged as completed.

Sprint retrospective: once the review is completed the team will reflect on the past sprint and identify possible improvements that will be tackled in future cycles.

Sprints

A sprint refers to a single iteration and incrementation cycle. The term 'sprint' is used because of the relatively short time period during which the activities will take place. Each development phase will typically last between one and four weeks. This speed of iteration can lead to high-quality, rapidly deployed systems. However, it must be carefully managed, and such a fast-paced environment can place a lot of pressure on developers.

INFORMATION FLOW DIAGRAMS

Information flow diagrams (IFDs) are visual representations that show how information travels within and between systems, organisations and processes. IFDs are useful for explaining complex systems and identifying where and how different parts of a system or process communicate. They allow developers to plan new IT solutions or identify inefficiencies in existing processes to improve the quality of information flow.

IFDs are made up of four main components:

Entities: an entity is a representation of sources and destinations of information. Entities can be anything that would receive or generate data such as customers, databases or external systems.

Flows: the direction that information will flow is shown using arrows that link the source and destination entities.

Processes: any transformations or processing of the information that occurs is shown in a box.

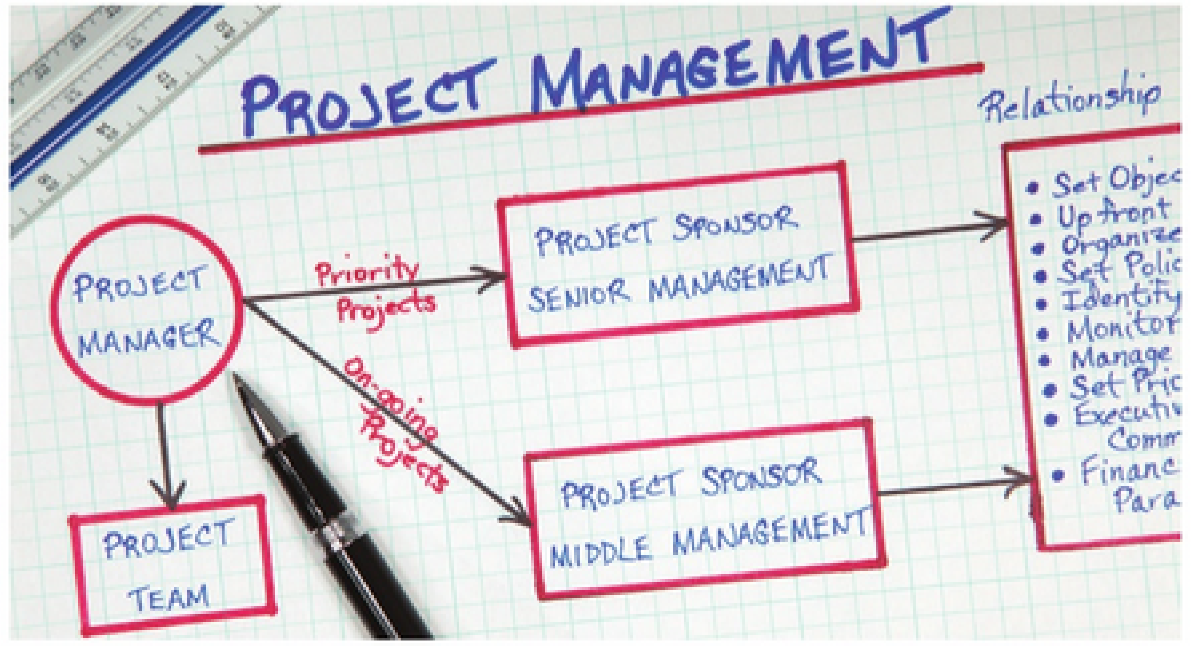
Decisions: as with a flowchart, any logical decision that must be applied is shown using diamonds. These depict where the flow of information is dependent on a specific condition.

IFDs are used to demonstrate information at different levels of depth and complexity. The three levels of IFD are:

Context or level 0 diagram: a high-level overview showing the entire system and its external interactions.

Level 1 diagram: a more detailed view of the information flow, focusing on major subsystems and information flows within them.

Level 2 diagram: a highly detailed view, exploring specific processes and how they handle information.


PROJECT MANAGEMENT
Relationship
PROJECT MANAGER
Priority Projects
On-going Projects
PROJECT SPONSOR SENIOR MANAGEMENT
PROJECT SPONSOR MIDDLE MANAGEMENT
PROJECT TEAM
• Set Objec
• Up-front
• Organize
• Set Polic
• Identify
• Monitor
• Manage
• Set Pric
• Executiv
Comm
• Financ
Para

▶ Information flow diagrams help to explain how systems and processes work as well as the different levels of complexity.

ACTIVITY 6

An online retailer needs to update their systems to improve efficiency and reduce costs. Before they do this, they need to have a thorough understanding of how information currently flows into, out of and within the organisation.

The organisation has customers and suppliers, and it uses third-party delivery companies as well as providing its own delivery services.

Create a Level 0 information flow diagram that could represent the organisation's communication. As well as the external entities (customers, suppliers and delivery companies), you will need to consider entities within the organisation, such as accounting, inventory (stock control), warehousing and IT.

Compare your information flow diagram with a classmate's diagram. Are your diagrams the same?

Discuss any differences and make any changes to your diagram if you think they are needed.

EXAM TIP

When responding to a diagram question, ensure that you use a ruler or similar to help you draw straight lines and clear symbols.

CHECKPOINT

1. Why is effective project management needed when developing systems?
2. Name three characteristics of a successful IT project.
3. What are SMART targets and why are they used?
4. Give two examples of SMART targets.
5. List three tools that are used to manage projects.
6. What is the waterfall development methodology?
7. Explain what is meant by the agile approach to development methodology.
8. Describe the three levels of information flow diagram.

SKILLS

CRITICAL THINKING, INTERPRETATION, REASONING

EXAM PRACTICE

1. Explain the meaning of T in SMART targets. [2 marks]
2. Draw a diagram to show the steps of the waterfall methodology. [4 marks]
3. Explain **two** benefits of using a Gantt chart when developing a project plan. [4 marks]
4. An IT development team are developing a smartphone app that will track the health and fitness of users. The team intend to use an agile project methodology when developing the app. Evaluate the suitability of agile methodology for this type of project. [12 marks]

TOPIC 16
EMERGING TECHNOLOGIES

INTERNET OF THINGS

Information technology is an area of study that is constantly evolving. As faster and more powerful computer processing becomes available to more people, the opportunity for technology to be used to enhance everyday life also increases. In this topic you will explore some new and emerging technologies and consider how and why these technologies can be used by individuals and organisations, as well as the impact that these technologies can have.

TOPIC 16 EMERGING TECHNOLOGIES

LEARNING OBJECTIVES

By the end of this topic you should be able to understand:

- the concepts, features and functions of machine learning
- the potential impact and possibilities of machine learning
- how and why augmented and virtual reality can be used by individuals and organisations
- how the internet of things (IoT) can affect individuals and organisations
- the technology and infrastructure that enables and supports the IoT
- potential security issues relating to the IoT
- how to produce high-level plans for IoT solutions.

GETTING STARTED

The image below shows some early mobile phones (cell phones) that were available in the late 1990s to 2000s. Compare these with the types of smartphones that are available today. How has the technology changed in such a relatively short time span? What features does your smartphone have that were not available a few years ago?

▲ First generation mobile phones look quite different from modern smartphones. How does your phone compare with these phones?

KEY TERM

artificial intelligence (AI): a computer science/IT discipline that aims to build computer systems that can replicate human intelligence

MACHINE LEARNING

Machine learning is a branch of **artificial intelligence (AI)** that allows computers to learn and improve without being explicitly programmed. In Topic 14 you learned about expert systems, which make decisions using a predesignated set of rules that look at the entered data and reach the same conclusion that an expert in the field would reach. However, an expert system can only operate in a very narrow scope, and anything that does not fit the pre-determined rules can be difficult to diagnose. **Machine learning** differs in that the decisions are not pre-programmed; instead a machine learning system should be able to identify trends and correlations in large amounts of data and then apply what it has 'learned', even if what it sees does not exactly fit a specific rule.

KEY TERMS

machine learning: a subset of artificial intelligence; the use of algorithms whose performance improves as they are exposed to more data

supervised learning: the use of labelled data to train a machine learning model

unsupervised learning: the use of unlabelled data to train a machine learning model

Machine learning involves training the system using large quantities of data so that the system can identify patterns, including ones that may be extremely difficult for a human to spot. The system can then make predictions about new data. Machine learning has a wide range of applications in various sectors and is becoming increasingly used in healthcare, finance, manufacturing and transportation to provide insights into data that may have previously been unnoticed, and to give a performative improvement or competitive edge.

CONCEPT, FEATURES AND FUNCTIONS OF MACHINE LEARNING

There are four main features that contribute to the overall structure of a machine learning system or solution: algorithms, data, training and predictions.

Algorithms

These are sets of instructions that the computer follows to process data and make decisions. In machine learning, algorithms help the system learn from data and improve their performance over time. These algorithms can be extremely complex, and they rely on mathematical and statistical processes. Many algorithms are now widely available in pre-programmed packages. These AI models, as they are referred to, allow companies to identify the type of problem that needs solving and choose a pre-written algorithm (or set of algorithms) to meet their requirements. The main types of problems that are solved by machine learning are the following:

Regression problems: a type of **supervised learning** used to predict continuous outcomes. It is used to establish a linear relationship between independent variables and a dependent variable by fitting a straight line to the data points. For example, predicting future sales based on historical sales data.

Classification problems: a type of supervised learning where the goal is to predict the categorical class labels of new instances based on past observations. For example, identifying spam emails by comparing the content of an email against other emails that have been labelled 'spam'.

Clustering problems: an **unsupervised learning** technique used to group similar data points together based on certain characteristics or features. For example, grouping customers based on purchase histories or demographic characteristics so that they can be targeted as a group for marketing, etc.

Anomaly detection: this can be used in both supervised and unsupervised learning models; it is a technique used to identify data points that fall outside of 'normal' behaviour. For example, identifying fraudulent transactions in the banking industry.

Data

Machine learning requires large amounts of data to work. One of the contributing factors to the growth of machine learning in recent years is the increasing amounts of data generated by computer systems. As you saw in Topic 12, computer systems continuously generate data and the amount of available data increases, as more and more devices interact and communicate. This data powers machine learning, and the quality and quantity of the data has a significant effect on the accuracy and effectiveness of the learning process.

Machine learning can be used with many different types of data, such as text, images, numbers and sounds. The data used and the type of problem will determine the model that is used.

Training

The 'learning' part of machine learning is referred to as training. This part of the process involves feeding data to the algorithm so that the system can learn from it. During training the data is typically split into two sets: the training set and the test set.

Training set

This is the larger portion of the data used to train the model. The model learns from the patterns and relationships in this data and will adjust its internal parameters as the model is run. This allows it to adjust the decisions it makes, which reduces the error rate in any predictions made. For example, when training a model to identify spam emails, the model will look at a given data point (such as the email text) and make a prediction as to how likely it is that the given data is spam; it will then compare its prediction to the actual outcome (label) and calculate the difference between the two based on the result. It will then adjust the parameters to inform the decision-making process.

Test set

This is a smaller portion of the data used to evaluate the model's performance after training. The model is not exposed to this data during training, and its predictions on this unseen data allow you to assess the quality and accuracy of the model. In the case of very complex problems, it may be necessary to refine the model and train and test over several iterations until you are satisfied that it performs as required.

Predictions

Once you are satisfied that the model is working as expected, it can be applied to new data to make predictions. For example, identifying cars and pedestrians in images from a self-driving car, or determining if a comment on social media is positive, negative or neutral.

Supervised and unsupervised learning

The way in which a machine learning model is trained can be broken down into two main approaches: supervised and unsupervised learning.

Supervised learning

Supervised learning uses a 'labelled' dataset to train the model. Each training example on the dataset contains an input, which can be a single datapoint or multiple datapoints, and a corresponding output (the actual category). The goal of supervised learning is to determine the relationship between the data and the label (this is known as a mapping function). This mapping allows the model to make more accurate predictions when presented with new, unseen data. Supervised learning is used where a problem can be categorised as either a regression or classification problem.

CASE STUDY: IRIS DATA SET

The iris data set is an early, well-known example of the use of data to solve a classification problem, using a labelled data set to determine which one of three classes an iris plant belongs to, based on provided characteristics.

The data set provides the input (four data points) and the corresponding output (label).

Here is an extract of the data set:

EXTRACT FROM THE IRIS DATA SET

Sepal length (cm)	Sepal width (cm)	Petal length (cm)	Petal width (cm)	Species
5.1	3.5	1.4	0.2	setosa
4.9	3.0	1.4	0.2	setosa
4.7	3.2	1.3	0.2	setosa
4.6	3.1	1.5	0.2	setosa
7.0	3.2	4.7	1.4	versicolor
6.4	3.2	4.5	1.5	versicolor
6.9	3.1	4.9	1.5	versicolor
5.5	2.3	4.0	1.3	versicolor
6.3	3.3	6.0	2.5	virginica
5.8	2.7	5.1	1.9	virginica
7.1	3.0	5.9	2.1	virginica
6.3	2.9	5.6	1.8	virginica

In a machine learning problem, the model would look for correlations between the four input values and use these to make a prediction for the class. The higher the correlation, the more accurate a prediction will be.

In the version of the full data set used in this example, these correlations were discovered:

CORRELATIONS FROM THE IRIS DATA SET

	Min	Max	Mean	SD	Class correlation	
Sepal length	4.30	7.9	5.84	0.83	0.7826	
Sepal width	2.00	4.4	3.02	0.43	−0.4194	
Petal length	1.00	6.9	3.76	1.76	0.949	High!
Petal width	0.10	2.5	1.20	0.76	0.9565	High!

What do you notice about the correlations? Which of the data points would provide a more accurate prediction of the class of the iris plant?

Unsupervised learning

Unsupervised learning uses 'unlabelled' or 'unknown' data to train the model. Unlike labelled data, the data has no corresponding output labels. Because the data is not labelled, the type of problems it is used for are different from those where labelled data is used. Rather than attempting to categorise the data, unsupervised learning tries to find patterns or structures in the data so that inferences and insights can be drawn, without any prior knowledge of what those patterns might be. This can be very useful for companies, as they can use it to find hidden patterns without influence from bias or assumptions from the user. Unsupervised learning is used where a problem can be categorised as either a clustering or a dimensionality reduction problem.

SKILLS

INTELLECTUAL INTEREST AND CURIOSITY, INTERPRETATION

ACTIVITY 1

Machine learning uses a range of different algorithms and models depending on the types of problem that need to be solved.

Investigate different machine learning models and how they are used. Use these lists to help you:

Models for regression problems:

- Linear regression
- Logistic regression
- ARIMA (autoregressive integrated moving average).

Models for classification problems:

- SVM (support vector machines)
- trees and forests
- Bayesian methods.

Models for clustering problems:

- K-Means
- PCA (principle components analysis).

Create a short presentation that explains each of the different models and their uses.

IMPACT AND POSSIBILITIES OF MACHINE LEARNING

Machine learning has a wide range of potential applications. In addition to the use of more conventional numerical data to classify objects, which you saw in the iris plant data example, machine learning provides interesting and powerful tools that can have a large impact on individuals and organisations.

Natural language processing

Natural language processing (NLP) focuses on how computers and humans can interact in a way that uses more regular everyday language. The main aim of NLP is to enable computers to understand, interpret and generate human language in a way that is both meaningful and useful. For example, where computers would traditionally try to process individual words from a user's input exactly, which could lead to inaccurate or unpredictable results, NLP also looks at context and has a better understanding of regional variations in language. NLP continues to have an impact on a number of areas.

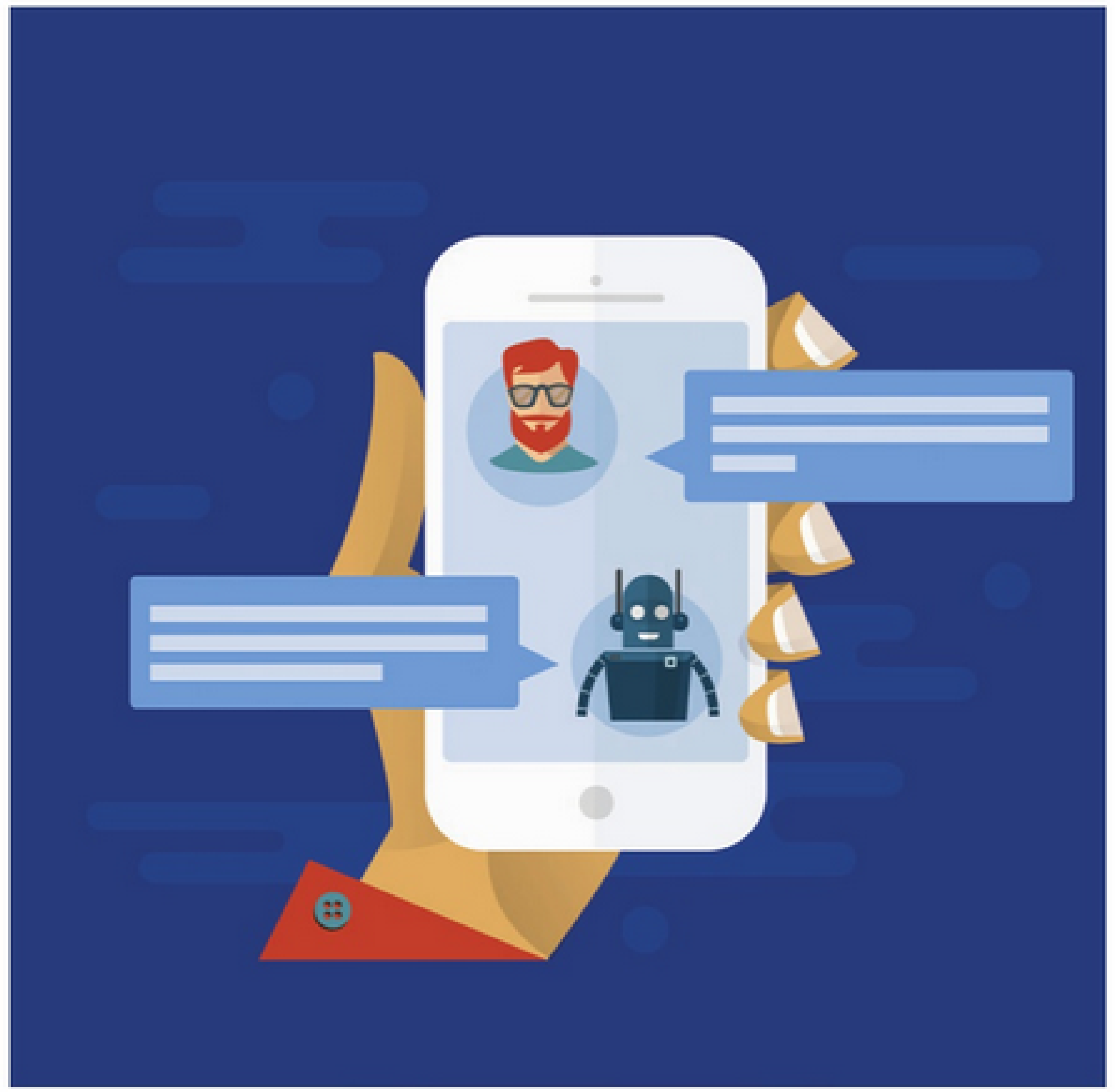
NLP in IT is not the same as NLP in Psychotherapy, which stands for Neuro Linguistic Programming.

Improved search engine results: NLP allows search engines to better understand a user's intent to provide more relevant results.

Virtual assistants: NLP, paired with speech recognition, allows virtual assistants, such as Siri and Amazon Alexa, to interpret and understand accents or regional variations in grammatical structure and word usage, as well as questions phrased in non-typical ways.

Automated language translation: NLP provides higher-quality results in language translation. Whereas early language translation tools tended to translate text literally, NLP allows these systems to draw on large sets of data to provide translations that are much closer to those of native speakers.

Chatbots for customer service: companies continue to explore how NLP can improve the quality and effectiveness of chatbots to improve the responsiveness and efficiency of these tools to support to customers, particularly for common or less complex queries. This frees human agents to deal with more complex or critical issues, resulting in better customer service.



▶ Machine learning aims to improve the quality of chatbots

Speech recognition

Speech recognition as a tool for interaction with computers has been used for a surprisingly long time, with commercial applications becoming available in the 1980s and 1990s. The initial focus for speech recognition was as an accessibility tool, or for limited automation such as a telephone-based helpline.

As computing power, access to data and complexity of algorithms have developed, speech recognition has become much more widespread. Early iterations of speech recognition software had to be trained by each user so that the system could store samples of the user's voice to use for comparison; this often took a long time and was often quite inaccurate. Now tools, such as Siri by Apple or Amazon's Alexa, use vast stores of data along with natural language processing to improve the quality and accuracy of how spoken language is processed.

▶ Do you have a smart speaker or similar device at home? How has it changed the way that you perform tasks and interact with devices?

Image recognition

Image recognition is a relatively new but quickly expanding area of machine learning. Because of the significant variation in the content of images, they are very difficult for computers to process and understand. Image recognition aims to harness machine learning to improve the accuracy of a computer when processing images. The technology has a lot of possible applications and presents opportunities for improving safety critical systems, informing professional decisions and improving the lives of those with additional needs. Some applications include the following:

Self-driving cars: image recognition is used to identify objects such as pedestrians, vehicles and traffic signs, and provide the onboard computer and driver with relevant information to help with decision-making.

Medical imaging: image recognition has been used to identify abnormalities in scans and X-rays that were too small for doctors to notice, enabling doctors to intervene earlier and provide better outcomes for patients.

Security and surveillance: image recognition is used to identify suspicious activity or individuals in video footage.

Social media: automatic tagging of photos of faces or objects is done using image recognition software.

Product search: image recognition enables faster and more accurate product searches by image.

Assistive technology: image recognition uses the camera of a smartphone to identify objects and provides audible descriptions of the objects for people with sight-loss.

SKILLS

ETHICS, REASONING/ ARGUMENTATION

ACTIVITY 2

Driver aids and self-driving cars rely on machine learning and image recognition to make decisions about the environment around them. However, their use is an area of much debate.

For example:

A driver has enabled the self-driving mode in their car. The image recognition system fails to identify a stop sign, which results in a crash that injures the occupants of another vehicle.

With a partner, discuss who you think is to blame for the accident. Was it the driver of the car that caused the accident, the manufacturer of the self-driving car or the individual who created the program that controls the car? Explain your point of view.

Pattern recognition

Pattern recognition is a concept that is used in all aspects of machine learning. It is the process of looking for trends, correlations and similarities in sets of data. In machine learning, the algorithms that make up the ML models are used to discover, learn and adapt to patterns in data, and can often exceed human capabilities with the scale and complexity of data that can be considered, allowing them to identify patterns that humans might miss.

VIRTUAL AND AUGMENTED REALITY

VIRTUAL REALITY

Virtual reality (VR) is a technology that provides users with an immersive experience within a computer-generated environment. VR aims to create a feeling that the user is actually present in the virtual world. The technology for VR continues to evolve, providing increasing levels of immersion and more realistic experiences. Most VR systems have several key components:

KEY TERM

stereoscopic: combining two images (still or moving) taken at slightly different angles to give the illusion of depth

Headsets: these devices are worn like goggles and they display the virtual world using **stereoscopic** screens that provide a three-dimensional view. These headsets often integrate headphones and head-tracking technology. This creates an immersive effect and responds to your head movements, making the environment feel more natural.

Software: specialised programs generate the visuals, sounds and haptic feedback (simulated touch) that make up the VR experience. Improvements in the processing power of personal devices has led to a surge in the use of VR for personal entertainment.

Controllers: handheld devices allow you to interact with the virtual world. In addition to buttons and thumb sticks that would be found on a standard controller, these controllers will also track your movements to simulate realistic actions such as picking up objects or manoeuvring through landscapes.

▶ How do you think VR technology will evolve in the future?

Uses of VR

The immersive nature of VR provides a range of possibilities beyond that of just entertainment. Here are just a few examples.

Education: VR can provide realistic and immersive experiences that allow students to explore subject content in ways that a traditional classroom would not provide. For example, exploring historical sites or exploring the depths of the ocean.

Training: VR allows users to develop skills that may have a certain degree of danger in a safe environment – for example, pilots practising emergency scenarios or doctors rehearsing complex surgeries.

Design and engineering: VR can provide simulated 3D models of buildings before they are built so that architects can identify design flaws and get feedback from others, which allows them to adjust the design before construction begins.

THINKING LIKE AN IT SPECIALIST

Research recent developments in virtual reality (VR). What interesting developments are being made? How is the hardware and software developing from what is covered in this section? If you worked for a company as an IT specialist, can you think of any uses of VR in the workplace?

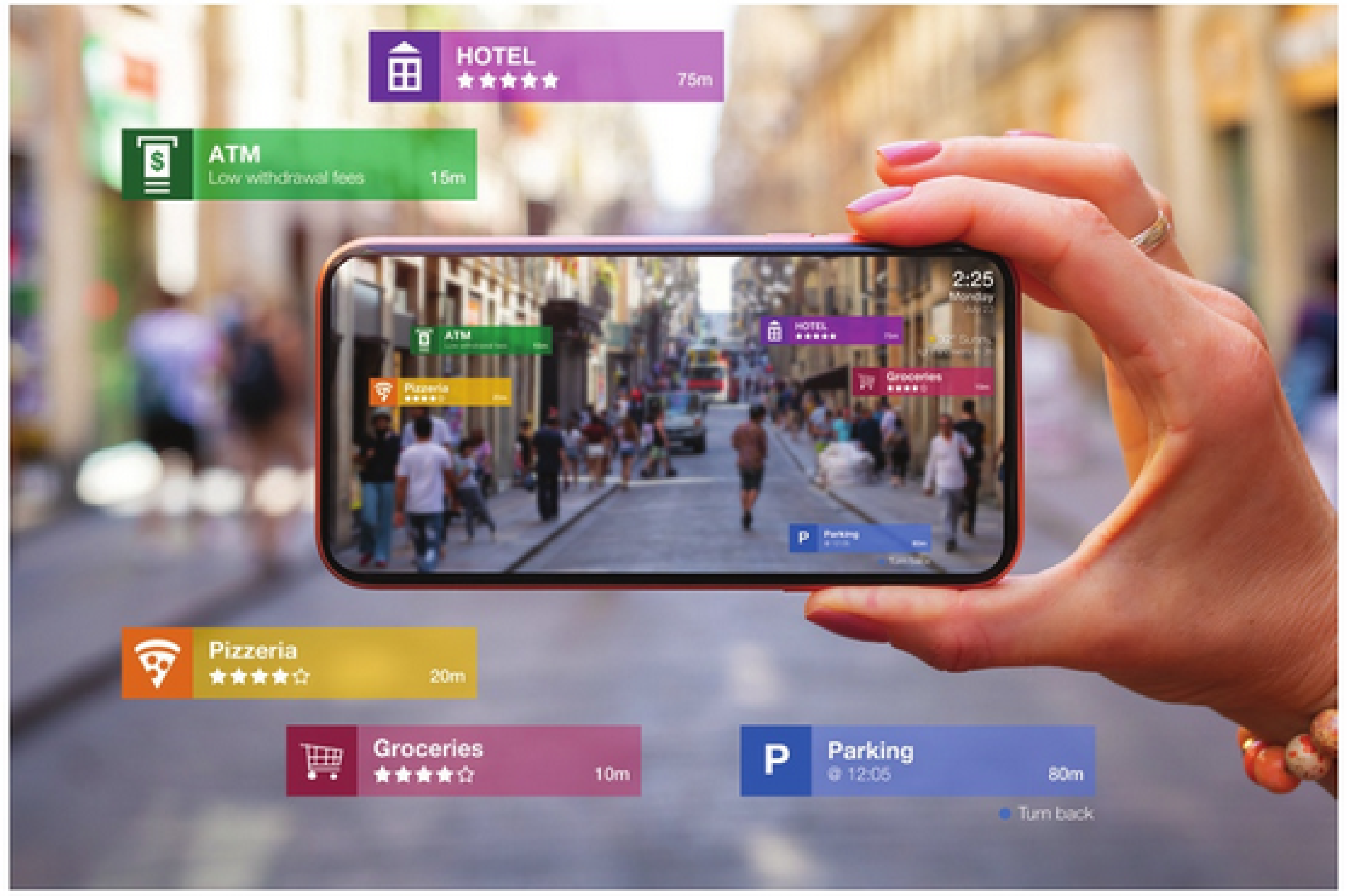
AUGMENTED REALITY

Unlike virtual reality, which aims to completely immerse a user in the simulated environment, augmented reality (AR) aims to blend the virtual and real worlds by superimposing computer-generated content on elements of the real world that are captured through a digital device. For example, some retailers provide an AR feature that allows you to select products, such as chairs or tables, and see how they look in your room before you buy them. Some other examples include:

Education: supporting the development of skills such as maintenance by adding virtual tags or labels to items to allow the user to more easily identify and learn the purpose of components.

Navigation: augmented maps that superimpose information and directions onto the buildings and roads as you explore.

Entertainment: interactive games where digital elements are blended with the real world, creating unique experiences.



▶ Augmented reality can provide additional information and context to real-world situations

CASE STUDY: POKÉMON GO

Released in 2016, the game Pokémon Go became very popular, due in no small part to its use of augmented reality. Although other companies had made use of AR prior to this, Nintendo reignited interest in AR and demonstrated to other companies what was possible.

Pokémon Go used the increasing power and capabilities of smartphones and a combination of map data, location services and the device's camera to create a virtual world on top of real-life locations. The game encouraged users to find Pokémon by exploring real-life locations. The requirement to move beyond the user's current location added variety and interest to the game.

Next steps:

Investigate augmented reality shopping apps that are available for your smartphone. For example, IKEA and Amazon apps include augmented 3D models to allow you to visualise items for sale in your own home.

INTERNET OF THINGS

The internet of things (IoT) is a network of devices that would not traditionally be considered 'smart' (such as household appliances), but which have been enabled to generate, send and receive data. For example, IoT devices like thermostats, lights and security cameras can be interconnected so that they can be controlled and monitored remotely via a smartphone app. In industrial settings, IoT sensors can be used to track inventory levels, or the location of a delivery in real time. There are three key concepts that drive the IoT:

Connecting devices: everyday objects are equipped with sensors that allow them to collect data related to their key function. The devices also need to be provided with some form of data connection so that they can share that data with other devices and systems.

Exchanging data: devices communicate with each other and with other systems. This can be within a limited/private network, such as a smart watch connecting directly to the user's phone, or using wider networks such as transmitting the data over the internet.

Automating tasks: the goal of the IoT is to make our lives easier and more efficient by automating routine or repetitive tasks. For example, systems that monitor the weather data and environmental factors and automatically water plants, or that send an alert to people to take hay fever medication when the pollen count is high.

IMPACT OF THE INTERNET OF THINGS

The IoT has had a significant impact on our daily lives; even if you do not have a 'smart home' with remote controlled lights and heating, you will encounter the IoT. For example, it is common for companies to provide information about the progress of an order. From the moment the order is placed, a system keeps track of when the order has been selected and packaged, when it has been dispatched and where the delivery van with the parcel is at any given point. All this data is then sent to the customer to track on an app or website. Interconnected systems generate a huge amount of data and information, which provides a number of benefits as well as challenges.

Benefits of the IoT

Convenience

The goal of the IoT is to provide additional data and information so that our needs are more easily met, where possible through automation. For example, you might have a coffee machine that communicates with your alarm clock to ensure that there is a freshly made drink waiting for you when you wake.

Efficiency

The automation of routine or mundane tasks allows people to focus on other more beneficial activities. Automating tasks can save individuals and organisations time, energy and resources.

Safety

Connected devices can monitor for potential hazards and alert you to dangers. This area of the IoT has a multitude of applications.

In industry, hazardous systems can be monitored remotely around the clock and, when issues are detected, alerts can be sent to key personnel, regardless of their location.

In the home, sensors can be used to monitor carbon monoxide and alert you if it is at an unsafe level.

Some ride hire/taxi firms (such as Uber) provide the intended route and live location data to both the driver and the passenger. This data can be shared with others, and can alert the passenger if the driver diverts from the specified route.

Challenges of the IoT

While IoT provides a lot of benefits, as with all IT systems it also presents a variety of challenges and concerns that individuals and organisations must consider to ensure that needs are met and potential risks are mitigated. These include security, privacy, standardisation and environmental concerns.

Security

Security is a significant concern with connected devices. Connecting devices to the internet raises security concerns and requires robust security measures. Security concerns and how to manage them for the IoT are covered in greater detail later in this topic.

Privacy

One of the key factors of the IoT is that data and information is shared between devices and systems. As more and more data is generated, and more services require access to that data, ensuring your privacy becomes an ever-increasing concern.

SKILLS

INTELLECTUAL INTEREST AND CURIOSITY, ETHICS, REASONING

ACTIVITY 3

Many services now exist that allow users to track their physical exercise, such as walking, running and cycling. These services often contain 'social' features that encourage users to share their activities with others, for example:

- alerting others when an activity has been completed
- sharing the route information of the activity
- comparing performance with other users in your area
- suggesting routes that others have shared.

What potential privacy and safety concerns might these features cause for users and the organisation?

How could the organisations that run these services, and how could individual users, ensure that the service is used in a safe manner while maintaining the benefits that the service provides?

Produce a set of notes to organise your thoughts. Spend some time discussing your ideas with a partner. Did they agree with you? Did they come up with other concerns and issues that you did not think of?

KEY TERM

application programming interface (API): a set of functions that enable two computer systems or programs to communicate and share data

Standardisation

Different devices and systems need to be able to communicate seamlessly. This can present significant challenges when devices or services are created by different companies. Efforts throughout the industry have been made to provide some way of standardising some aspects of the systems, such as the types of connections that devices use to communicate (e.g. Bluetooth®, Wi-Fi, USB). For other aspects it is not always possible to standardise systems; for example, each system will be programmed in its own way, but will still need to communicate with a range of potential unknown systems. This interconnectivity has led to the increased use of a type of software known as **application programming interface (API)** as a standard way to exchange data between systems. An API is essentially a set of rules and protocols that tell a system how to communicate with another system.

Environmental concerns

Environmental impact is another big concern of the IoT. Environmental concerns are wide ranging and present individuals and organisations with a number of challenges. We look at some of these here.

KEY TERM

e-waste: discarded electrical equipment such as computers, smartphones and televisions

Disposing of **e-waste** improperly can lead to toxic substances leaking into the environment, especially if components are not separated for recycling. Many countries now have strict rules in place for the disposal of electronic waste. The disposal and management of e-waste is also affected by the lifespan of IoT devices. Many have short lifespans because of built-in obsolescence, software incompatibility or consumer desire to own the latest gadget. This commercial and social pressure that makes technology fashionable has led to increased e-waste generation.

Many IoT devices require continuous internet connection, increasing overall energy demand – the need to regularly recharge devices and maintaining constant power to the systems that support the infrastructure has led to significant increases in energy consumption.

IoT generates and uses vast amounts of data, which is growing exponentially. This data must be processed and stored. This usually occurs in large data centres, which contain large numbers of servers and storage units, and take vast amounts of energy to power the processors and storage and to keep the devices cool.

Resource extraction is also a challenge as manufacturing countless small devices requires raw materials to be mined, including rare earth metals. This contributes to resource depletion and environmental damage. Ensuring that any damage is reduced and that, where possible, resources are reused and recycled, is a significant issue.

SKILLS

ETHICS, COLLABORATION

ACTIVITY 4

Some critical minerals for IoT devices come from conflict-ridden regions, raising concerns about ethical sourcing and human rights violations.

With a partner, research and discuss the impact that the IT industry can have on these regions and the people who live there.

Pollution is a major concern for many aspects of IT, including the IoT. In addition to the extraction of resources, the production of electronics can involve harsh chemicals and emissions that can lead to air and water pollution. Shipping large quantities of materials for manufacture between countries, and transporting devices from their point of manufacture to distribution centres and customers, contributes to greenhouse gas emissions, so increasing the problem for climate change and global warming.

SKILLS

INTELLECTUAL INTEREST AND CURIOSITY, PROBLEM SOLVING

ACTIVITY 5

Explore the environmental impact of the IoT in greater detail.

What systems and processes can individuals and organisations put in place to reduce the environmental impact caused by the IoT?

Produce a written report outlining your findings.

INFRASTRUCTURE FOR THE INTERNET OF THINGS (IoT)

The internet of things is an interconnected system of devices and everyday objects to share data. IT infrastructure is needed in order to make that happen, as with any IT system. That infrastructure can be broadly grouped onto four main areas: sensors, networks, embedded systems and storage.

Sensors

Sensors are devices that monitor conditions in the environment or in relation to the objects they are attached to, and convert these into a digital signal, which in turn becomes usable data. Sensors are the foundational components of IoT infrastructure – they are responsible for capturing real-world data such as temperature, humidity and motion. For example, smart watches and fitness trackers can detect heart rate, specific movements and activities, and in many cases geographical location.

These sensors can range from simple devices (such as a temperature sensor in a home) to more advanced ones with multiple sensing capabilities (such as biometric sensors that can collect fingerprints and monitor heart rates).

Sensors are the core component of the internet of things; it is their capabilities that allow us to convert everyday objects into smart devices that can provide a rich and endless supply of data.

SKILLS

CONTINUOUS LEARNING, INTELLECTUAL INTEREST AND CURIOSITY

ACTIVITY 6

Investigate different types of sensors and how they collect and use data.

Use this list of sensors that you would find in a smartphone to get you started:

- GPS receiver
- thermistor
- accelerometer
- gyroscope
- fingerprint senor
- microphone.

What other types of sensor did you find out about?

KEY TERM

4G/5G: fourth and fifth generation technology standard for cellular networks

Networks

The increase in availability and performance of networks in recent years has fuelled the adoption of IoT systems into everyday life. IoT systems are reliant on networks to facilitate the transmission of data between devices, sensors and other components, and they often require near constant connection to be effective. With the increased availability of broadband networks in the home, open public networks and cellular networks like **4G** LTE and emerging **5G**, it is much more convenient to harness IoT systems to automate solutions and provide insights into a range of areas of our lives. These networks can be wired or wireless, depending on the application requirements, and the choice of network technology depends on factors such as range, data bandwidth, power consumption and scalability.

Embedded systems

Embedded systems are specialised computing devices that are integrated into IoT endpoints. These systems typically combine a microprocessor or microcontroller, memory and input/output peripherals such as sensors, actuators and communication modules. An embedded system differs from a smart device in that it is usually designed to conduct a single or narrow scope of tasks, for example, a washing machine or fitness tracker. It is these types of systems that can essentially be thought of as the 'things' in the IoT. Embedded systems are designed so that they are compact, power efficient and operate with minimal computing resources. Because of their limited power and processing needs, and the way the software/firmware is installed on the system, they can also be extremely reliable.

Storage

Storage systems in IoT infrastructure are responsible for storing and managing the vast amount of data generated by sensors and devices. The data generated by IoT devices can vary significantly in terms of type, quantity and size. It includes the following types of data:

- Sensor data, such as temperature, pressure, humidity, location, sound, vibration, etc.
- Multimedia data, such as images from cameras or sound recordings from monitoring devices
- Log data, which can include service status, operational information and error messages
- User-generated data, which can include preferences, settings and even interactions with the data of other users.

Overall, the IoT consists of millions of devices generating continuous data streams. This requires massive storage capacity. Depending on the application requirements, data storage solutions can vary from local storage on the individual devices themselves to cloud-based storage services. In the structure of the IoT there are four main categories for storage areas:

- Edge storage: data is stored and processed on the device itself or nearby gateways before sending to the cloud. This is suitable for real-time applications such as tracking your activity during a run. This also helps reduce network congestion as data can be sent to alternative storage at less critical times.
- Cloud storage: data is stored in centralised cloud platforms, offering scalability, flexibility and accessibility from anywhere.
- Fog computing: data is processed and stored in distributed mini-data centres that are geographically closer to the devices, offering a balance between edge and cloud storage.
- Hybrid storage: combines edge, cloud and fog storage. This will typically be used by individuals and organisations that are attempting to achieve a balance between performance, security and cost considerations.

SECURITY ISSUES RELATING TO THE IoT

The interconnected nature of devices in the IoT, and the vast amount of sensitive data these devices generate and transmit, means that the security and integrity of the systems and data is a significant concern for the IT sector. When designing and using IoT systems, the risks and rewards must be carefully considered, and any potential issue mitigated to ensure the integrity, confidentiality and availability of data and information.

Privacy concerns

IoT devices often collect sensitive personal data, such as location information, health metrics and behavioural patterns. Unauthorised access to this data can lead to privacy violations, identity theft and even personal safety concerns.

Protecting **data privacy** can involve using strong encryption and access controls to ensure that data is not accessed by unauthorised individuals.

KEY TERM

data privacy: the protection of data from unauthorised access. Ensures that only those who have permission and a valid need to see the data are able to access it

Data integrity

Manipulation or tampering of data generated by IoT devices can have serious consequences. For example, a false reading in an industrial system could have catastrophic effects on the process or business. Incorrect health monitoring data could cause underlying issues to go unnoticed, resulting in further health concerns at a later date. It is important to implement measures such as data validation, to check whether data is within expected parameters, and integrity checking to detect any possible error, corruption or tampering, in order to help mitigate these risks.

Unauthorised access

IoT devices can be targets for unauthorised access by **threat actors** seeking to exploit vulnerabilities for various purposes, including data theft, sabotage or espionage. Weak authentication mechanisms, default passwords and unpatched software are common ways that threat actors attack systems. However, implementing strong authentication, regular software updates and security patches can help prevent unauthorised access and protect against previously undetected vulnerabilities or emerging threats. One of the key concerns with IoT is that, because of the nature of the devices, many embedded systems never receive firmware updates, which can leave vulnerabilities in a system.

KEY TERM

threat actors: also known as malicious actors or cyberthreat actors. Individuals or groups that intentionally target digital systems to cause damage, disruption or for financial gain

Denial of service (DoS) attacks

A DoS attack is a security concern for all networked digital systems, including IoT solutions. A DoS attack aims to disrupt the operation of systems or even cause service outages by overwhelming devices or networks with excessive traffic. Because of their heavy reliance on communication channels, IT systems are particularly vulnerable to this type of attack. DoS attacks can be reduced by implementing network segmentation (separating secure parts of the network from the less-secure parts exposed to the internet), traffic filtering (restricting the types of data packets allowed through a router or firewall) and rate limiting (restricting the number of data packets allowed).

Insecure communication channels

IoT devices often communicate over insecure channels such as unencrypted public Wi-Fi or Bluetooth® connections, leaving data vulnerable to interception and eavesdropping. Implementing encryption protocols such as SSL/TLS or IPsec can secure data in transit and protect it from unauthorised interception. You can also help protect your data by avoiding connection to open networks, such as those in shopping centres, hotels or cafes.

DESIGNING AN IoT SOLUTION

For an IoT solution, you need to consider what needs to be achieved and how the tasks will be implemented in order to generate the data and information. You also need to consider the individual components and systems that will be involved, and the processes and tasks that will be required. You should write down what you want and need (written plan) and then produce an information flow diagram to illustrate your design.

Example – smart home

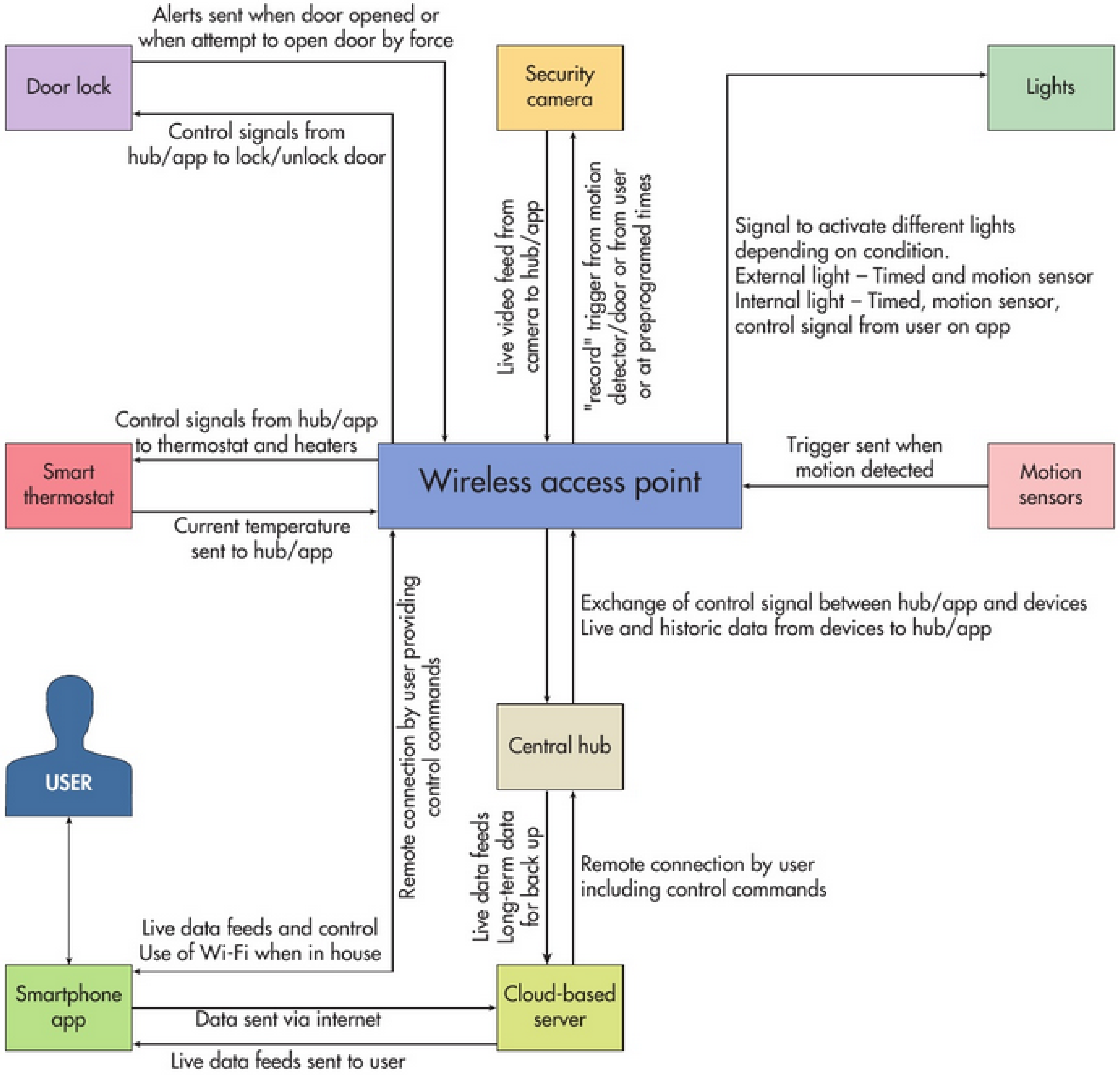
Written plan

Components needed: smart thermostats, door locks, security cameras, lights and motion sensors.

Description: devices will communicate with a central hub using wireless connection such as Wi-Fi. The hub connects to the internet and a smartphone app, allowing users to monitor and control devices remotely. The Wi-Fi connection will be secure and require a passkey to access. Long-term data is backed up on the internet using the subscription service from the device manufacturer. Short-term data is stored on the hub and on user devices. Long-term data can be accessed on the hub or in the cloud for analytics and historical tracking.

Diagram

See the information flow diagram in Figure 16.1.


Alerts sent when door opened or when attempt to open door by force
Door lock
Security camera
Lights
Control signals from hub/app to lock/unlock door
Live video feed from camera to hub/app
"record" trigger from motion detector/door or from user or at preprogramed times
Signal to activate different lights depending on condition.
External light – Timed and motion sensor
Internal light – Timed, motion sensor, control signal from user on app
Control signals from hub/app to thermostat and heaters
Smart thermostat
Wireless access point
Trigger sent when motion detected
Motion sensors
Current temperature sent to hub/app
Remote connection by user providing control commands
Exchange of control signal between hub/app and devices
Live and historic data from devices to hub/app
USER
Central hub
Live data feeds
Long-term data for back up
Remote connection by user including control commands
Live data feeds and control
Use of Wi-Fi when in house
Smartphone app
Data sent via internet
Cloud-based server
Live data feeds sent to user

▲ Figure 16.1 Information flow diagram for a smart home

SKILLS

INTERPRETATION, ANALYSIS, CRITICAL THINKING

ACTIVITY 7

Smart agriculture system

Written Plan

Components needed: soil moisture sensors, weather stations, drones, GPS trackers and automated irrigation systems.

Description: sensors will be placed in fields to collect data on soil moisture, temperature, humidity and weather conditions. This data will be transmitted to a central management system, where monitoring software will analyse data in order to optimise irrigation schedules, monitor crop health and detect pests or diseases. Drones equipped with cameras, sensors and delivery systems can provide aerial imagery for crop monitoring and analysis, and where needed deliver additional nutrients and pesticides.

Draw an information flow diagram (IFD) for the smart agriculture system outlined in the written plan.

CHECKPOINT

1. What is machine learning and how can it be used by organisations?
2. Explain how data is used to train a machine learning model.
3. What is augmented and virtual reality and how can it be used by individuals and organisations?
4. What is the IoT?
5. Describe an IoT solution that could be used by an individual.
6. Describe an IoT solution that could be used by an organisation.
7. Explain three components of the infrastructure that enables and supports the IoT.
8. What are the potential security issues relating to the IoT?
9. What is a high-level plan and how is it used when designing IoT solutions?

EXAM TIP

Although the specification does not dictate specific symbols for use in an IFD, you should follow common conventions (e.g. rectangles for entities, arrows for data flow) to ensure that your response is clear and easy to understand.

SKILLS

CRITICAL THINKING, INTERPRETATION

EXAM PRACTICE

1. Describe the process of training a machine learning model using supervised learning. [6 marks]
2. Explain two applications of natural language processing. [4 marks]
3. Discuss the positive and negative impacts that the use of AR and VR can have on the education sector. [6 marks]
4. Dawid is implementing a healthcare monitoring system that will make use of the internet of things (IoT). He has produced a written plan for the system:

   Components needed: wearable devices (e.g. fitness trackers, smartwatches), medical sensors (e.g. heart rate monitors, blood glucose meters) and remote patient monitoring systems.

   Description: wearable devices and medical sensors will be distributed to patients and will continuously collect biometric data. The data from the device/sensors will be sent to a custom app on the user's smartphone via Bluetooth®. Collected data will be transmitted to a healthcare provider's platform via the internet using any suitable connection. Machine learning algorithms analyse the data to detect anomalies and provide early warnings for potential health issues. Healthcare professionals can remotely monitor patients' health status and intervene as needed. Users can monitor their own data using the app.

   Produce a high-level diagram of Dawid's planned system. [9 marks]

TOPIC 17
USES AND FEATURES OF DATABASE SOLUTIONS

Databases are at the heart of many IT applications from banking to social media to online shopping. In this topic you will look at how and why databases are used for a variety of purposes. Applications that use database technology are often developed by IT professionals on behalf of the users, so you will also look at how the needs of those users and the database features required are identified. Users of course need to interact with the database application so you will consider how you can make the user experience of the application a positive one.

TOPIC 17 USES AND FEATURES OF DATABASE SOLUTIONS

LEARNING OBJECTIVES

By the end of this chapter you should be able to understand:

- why databases are used and the typical applications they are used for
- the features of database applications
- the different user needs including how a database can be evaluated
- how the database application can impact on the user experience
- how to understand the characteristics of the user and evaluate their experience.

GETTING STARTED

Can you think of any examples of where database technology is used? Are there any databases you use regularly (don't forget database technology is used to support many web-based applications such as social media and online shopping)? How does a database differ from a simple list? What features of databases make them so widely used?

▶ Computer databases replace vast quantities of traditional paperwork

DATABASE APPLICATIONS

WHY ARE DATABASES USED?

Almost every IT application needs to store data. While a program is running, that data is stored in memory using variables, but for the data to persist after the program ends it needs to be written to a file stored on disk. You will probably have written a program that writes some data to a text file, but that will probably have been a fairly small amount of data with very little structure to it. When dealing with a large amount of data it needs to have a structure to make it useful. As you will have learned, this is one of the key differences between data and information. The most important method used to structure data is to divide it into records and fields.

Suppose you had data about a football team's matches and results; the data might look something like this:

High Town Rovers, 2/12/23, home, 2, 1, Northern Athletic, 19/12/23, away, 0, 1

While you can probably guess what this data means, it makes more sense (certainly to human users) if you structure this data into records and fields as shown in Table 17.1.

TABLE 17.1: AN EXAMPLE RECORD STRUCTURE

Record no.	Opponent	Date of match	Home or away	Our score	Opponent score
1	High Town Rovers	2/12/23	Home	2	1
2	Northern Athletic	19/12/23	Away	0	1

This data could easily be recorded using a spreadsheet, and for many applications that deal with fairly small amounts of data, a spreadsheet is ideal. In terms of storing data, the main differences between spreadsheet and database software are as follows.

Spreadsheets are easier to set up and understand than databases and many users will be familiar with spreadsheet software.

Spreadsheets only support 'flat' files and do not support creating relationships between different data files. Data relationships are covered in detail in Topic 18, pages 00–00.

Database software is optimised to support large datasets with millions of records, which would not be appropriate for spreadsheet software.

Database software typically includes sophisticated tools for data management and manipulation, such as tools to search and sort data, which provide more features than those provided in spreadsheet software.

Database software typically provides tools for creating end user applications including forms to allow data input (and output) and reports to provide formatted output of relevant data.

Spreadsheets are focused on single user access and although a spreadsheet can be shared between several users, most database software is optimised for multi-user access.

In general, spreadsheets are more focused on providing end users with features that allow them to manipulate small amounts of data, whereas database software is focused more on providing tools to allow IT professionals to build applications to store and manipulate large quantities of data, which can then be used by end users.

Some of the applications that databases are used for include:

Stock control – many organisations in retail, distribution and manufacturing need to keep a stock of products, components, raw materials and so on. A database system is commonly used to keep track of how many of each item is in stock and also manage the reordering of items when the stock falls below a certain level.

Ticketing – selling tickets for flights and theatres, as well as booking applications for hotels and holidays are all candidates for a database solution. It is particularly useful to have a central database of anything that can be accessed from different locations, so that bookings can be coordinated. This can include available and booked seats, rooms, tickets and so on.

KEY TERMS

relational database: a database made up of multiple related tables

transaction: a sale, where a customer exchanges money for goods or services

Banking – modern banking would be impossible without database software to store the records of millions of customer accounts and keep them up to date with **transactions** that could take place anywhere in the world.

E-commerce – this is another application that could not function without database software. An online store needs a database of the products it has for sale, with all the details, images, reviews, and so on included. Stock details are also typically included so customers can see if items are in stock or not. Databases also store details of registered customers and of sales that have been made.

Scientific research – data is very important for scientific research because it allows changes and trends to be identified. Tracking and analysing changes in global weather patterns linked to global warming, for example, requires scientists to have access to data about weather (such as temperature, rainfall etc.) from many different locations over many years.

Databases are also used for many other applications not listed above, and are the primary method used to support many business applications.

DATABASE FEATURES AND USER REQUIREMENTS

Some database features are universal, in other words, they are required by every application. However, other features that a database may require are relevant to only certain types of applications.

Just about every application requires a database to have the following features:

the ability to store large amounts of data, while maintaining integrity and security

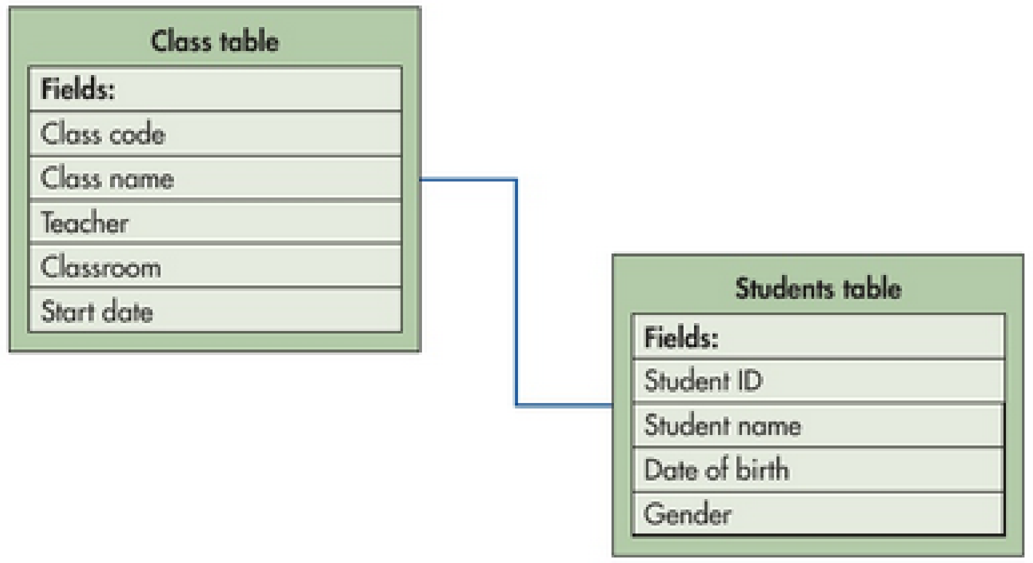
the ability to minimise data duplication so there is only one copy of each data item in the database

provision of tools to add, modify and delete data, plus tools to allow data to be searched for, sorted, filtered and manipulated in a variety of ways.

Avoiding data duplication is a key requirement of a database system and the design of a specific database. This is because if a data item is duplicated (for example, if a customer address is recorded in more than one place in the database) not only does that waste space, but more importantly, it introduces the possibility that if the data item is modified then it may not be modified in all the copies that exist. Therefore, for an efficient, accurate database, data duplication needs to be avoided.

The use of a database will determine its features, but you will need to consider the following:

The data that needs to be stored. What type of data and what fields will be needed? The IT professional developing the database will need to collect this information from users. Structuring the database into different related tables is something that is done at the design stage by an IT professional, but users need to describe what data needs to be stored. The subject of **relational database** design is covered in detail in the next topic, but the need to split data into different tables is a common requirement. For example, a database to record the different classes in a college or school would need fields to record things such as the name of the class, the teacher, the classroom where the class is held and so on. However, a single table could not store all the details of the students enrolled in the class, because each class is likely to have multiple students enrolled. This would require multiple records in a separate (but related) table. Figure 17.1 shows an example of linked tables.



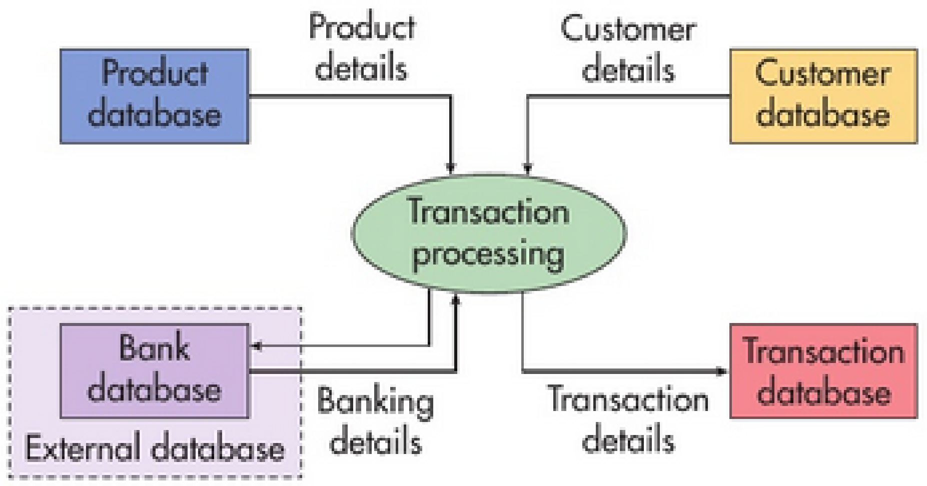
▶ Figure 17.1 Related tables in a database

The user interface. How will users interact with the database? It might be that the user interface is separate from the database itself. For example, with an online shop the user interface is provided through a website that will be designed and developed by web developers, not by the database developers, although they will need to work together to make sure the database can supply the data that the web pages require. In other situations, where users will interact directly with the application, screen forms and reports designed for printing will be required.

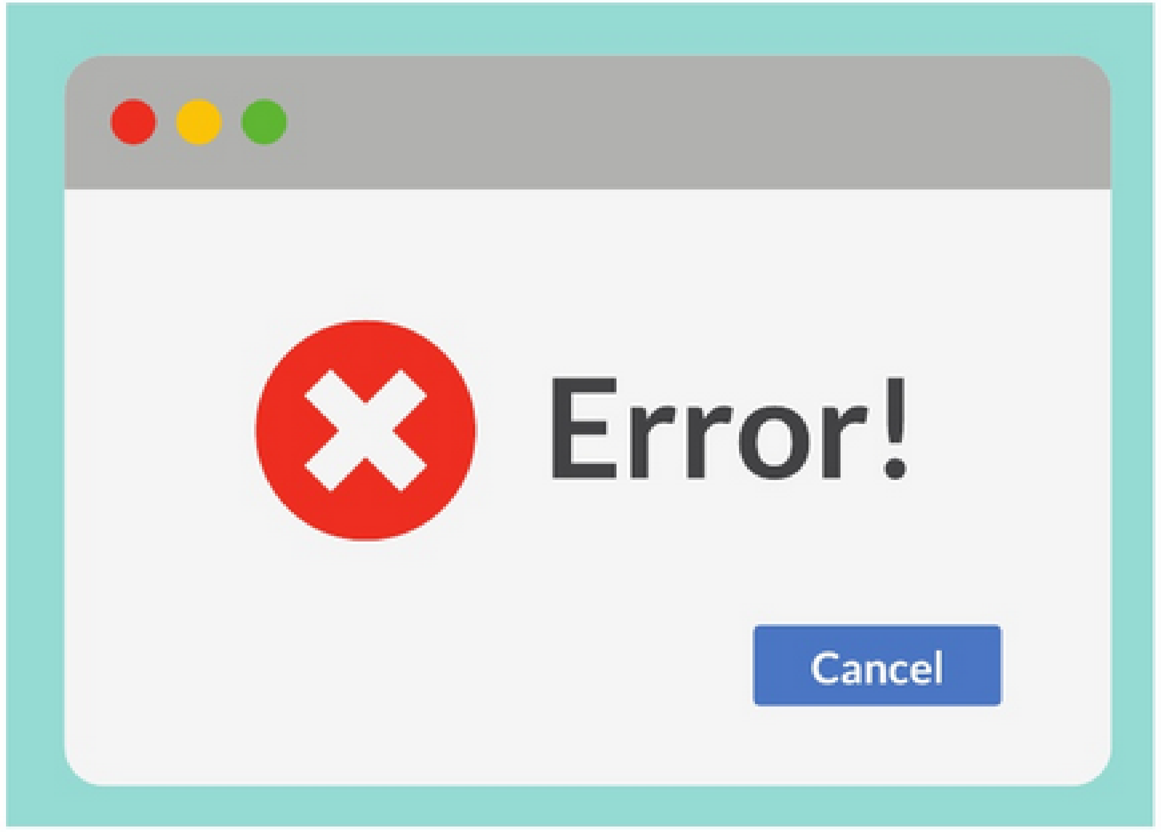
The processing involved. Databases don't just store data, they also manipulate and combine that data to meet the organisational requirements. The processing can involve:

- **Finding and selecting data.** Many database applications will have a requirement to find one or more records that match certain criteria. For example, a database that supports airline ticketing will need to be able to find free seats that match a customer's requirements (flight departure and destination airports, date and time). Searching and selecting data is done using database queries, which are described in more detail in the next section.
- **Calculations with data.** Database applications often need to carry out calculations with data to produce new or additional data. Some examples include:
  - A database that stores customer orders may need to add up the prices of the items on the order to work out the total cost of the order.
  - Databases used to support management decision making or scientific research may calculate averages and other statistics from the data they have stored to help identify trends and draw comparisons.
  - An organisation involved in selling goods through various retail outlets might want the database to produce reports to total up the sales from the different outlets over different periods (months, years, etc.) to compare the performance of different outlets and identify trends (such as rising or falling sales, profit, etc.).

Transactions. Transaction processing is done using databases. A transaction is essentially a sale, and typically involves at least three parties – usually the customer, the seller and the bank. Depending on the type of transaction, each of these is represented in a database and further database records are created by combining data from each of these parties. Take an online transaction for an example. The organisation selling the product will have a database of products for sale. They will also have (or create) a database record for the customer. When the sale is processed, the seller will get approval from the bank to transfer money from the customer's account to their account. The transaction will take data about the customer and the product and combine these to create a record on the transaction database. Figure 17.2 shows the interaction involved in processing such a transaction.



▶ Figure 17.2 Database interaction in recording a transaction



▲ Evaluating something means that improvements can be made

EVALUATING DATABASE FEATURES

An evaluation is about making a judgement as to how well a product matches what is actually needed. How well does it do the job it is supposed to do? Evaluations can be useful as they help people identify what things worked well and what things did not work well, so that improvements can be made. Evaluations also help people understand how to do a better job next time they approach a similar problem.

For you to be able to effectively evaluate a database and its features, you need to have a clear idea of what the user requirements are. The best way to approach this would probably be to create a list of the required features and then review each one against the actual database, considering the advantages and disadvantages of each feature, as well as any issues or problems. Questions you need to consider include:

Does the design of the database support the required application? Does it have appropriate tables, records and fields to allow the required data to be stored?

Does the database have the required features and do they work correctly? For example, a database designed to support stock control would need to have features to allow delivery of new stock to be added to the table(s) that record existing stock. It would also need to have features to allow items that have been sold to be removed from stock. The database should have been tested before the evaluation takes place so you should not be repeating the testing, but it may be that the requirements have been misinterpreted in some way or required features are missing.

Is the performance of the features acceptable? For example, do they produce results in a reasonable timeframe? Think about a system designed to allow you to look up customer details to answer a telephone enquiry. The system would need to be able to find the customer's record within a few seconds. Waiting anything longer than about 10 seconds to retrieve a record would not really be acceptable for an interactive system like this.

KEY TERM

intuitive: an interface that can be navigated by instinct and does not require instruction

How is the user experience? Is it clear how the features work and how to use them? Are all the menus and options clear and **intuitive** (i.e. it is obvious how to use them)? Is the output provided by the features as expected? For example, if the database produces a report does the report contain all the information required? There are many database features you can evaluate under the heading of 'user experience'. These are covered in the next section.

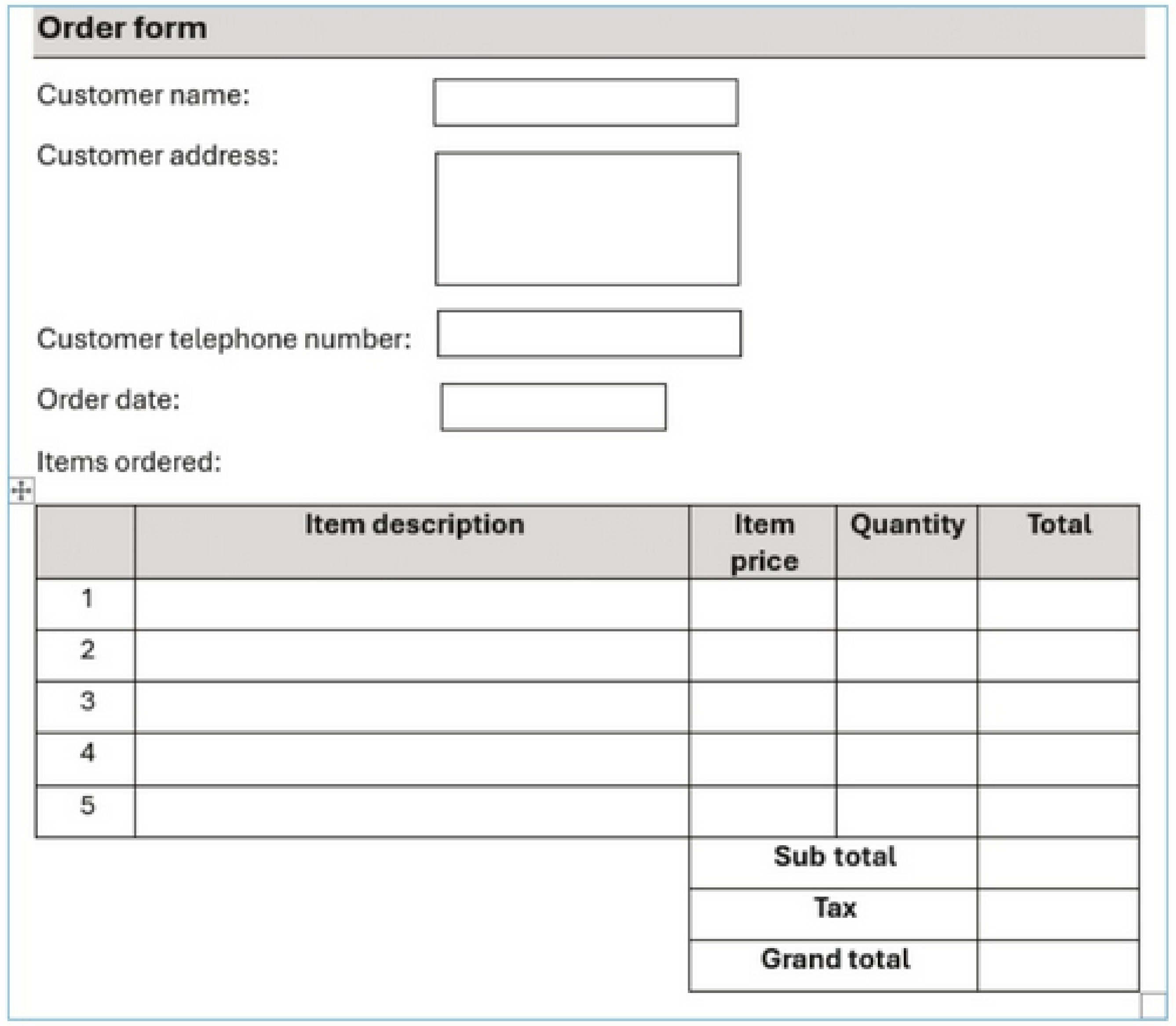
When writing an evaluation you need to be specific – it isn't very helpful to say things like 'I didn't like this feature', instead you should say 'I didn't like this feature because it …' and give clear reasons why you do or don't like a particular feature. Where possible, you should suggest improvements that will deal with any issues you have identified.

SKILLS

ANALYSIS, PROBLEM SOLVING

ACTIVITY 1

1. Figure 17.3 shows an example of a paper-based order form.



Order form

Customer name:

Customer address:

Customer telephone number:

Order date:

Items ordered:

	Item description	Item price	Quantity	Total
1				
2				
3				
4				
5				
		Sub total		
		Tax		
		Grand total		

▲ Figure 17.3 A paper-based order form

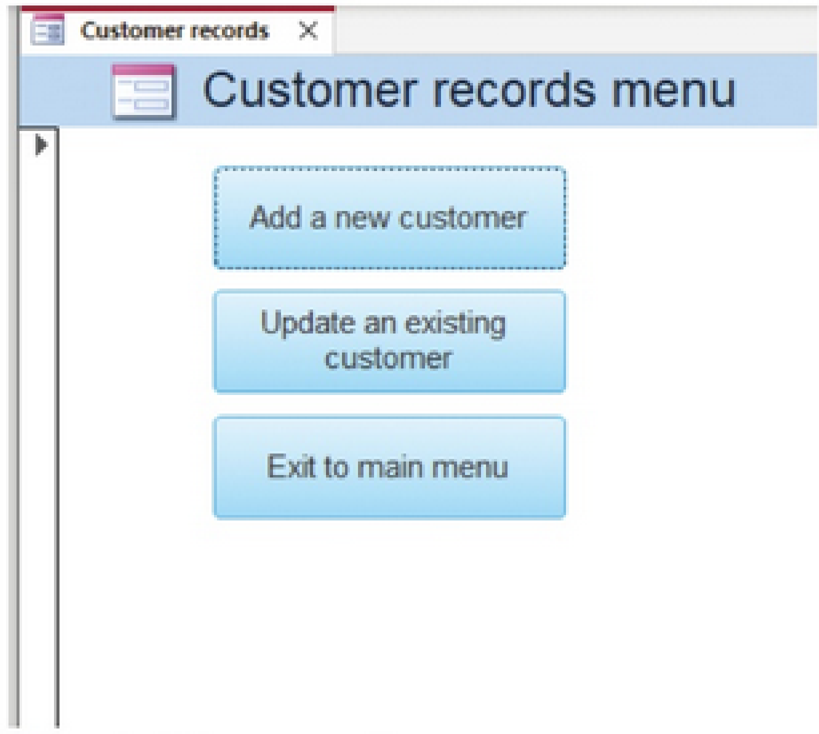
The IT professional developing a database to replace this paper-based method of recording orders has produced a draft record structure shown below.

Field name
Customer name
Customer address
Customer telephone number
Order date
Item description
Item price
Quantity
Sub-total
Tax
Grand total

Evaluate this structure in terms of its ability to support recording order details. Are the fields listed adequate to store all the details of orders? Are there any fields that are not necessary? Are there any issues with this list of fields?

2. Imagine a database used to book seats on flights at an airline. What basic database tables do you think would be necessary to support this kind of application?

3. Figure 17.4 shows a menu for a database system that allows a user to maintain customer records. Are there any obvious features missing from this menu?



▲ Figure 17.4 Customer records menu

USER NEEDS

A database application will typically be developed by an IT professional on behalf of a client or user. This is the situation in which most software is developed, and the same process for understanding user requirements that was described in Topic 15 can be used.

ANALYSIS OF USER NEEDS

Understanding the needs of the user is key to developing any successful IT application. The methods that can be used to collect information about user needs include the following:

Observation – one way of finding out the user requirements of a database is to observe them using the current system that the database will replace. This might be a paper-based system or a system that uses some other forms of IT, for example a spreadsheet. Observing people at work can be time consuming, especially if some tasks that need to be included in the system don't happen very often. Detailed notes should be made of the observation so that the user needs are fully recorded for later analysis.

Document analysis – if the database is replacing a paper-based system, then one way you can start to understand the requirements is to look at the paperwork (e.g. forms) that are currently used. This method can also be used if the existing system is IT based as you can look at what is currently in place, such as a spreadsheet or word-processed forms and documents.

Interviews – interviewing users is another method that can be used to understand their requirements. This method allows you to ask questions to make sure that you fully understand the process that needs to be automated.

In reality, one single method is unlikely to be sufficient, but a combination of these three methods is likely to produce the most accurate collection of user requirements. For example, some initial observations and document analysis could be backed up by interviews to clarify any issues and fill in any gaps in the understanding of the requirements.

USER CHARACTERISTICS

Not only is it important that the database you develop has the functionality required, it also needs to be appropriate for the users who will interact with the system. In some cases, you may not know exactly who the users are.

If your database will have a web front end (such as an online shop), you may not know exactly who your users are (because they can be anyone on the internet) but you may have an idea about the sort of user the website might attract. This is usually referred to as the website **target audience**. One of the key characteristics to consider about your target audience will be their degree of IT literacy. The user interface you provide will need to take this into account. If users have limited IT skills, the interface will need to be simple enough for them to use, and provide detailed instructions and help.

KEY TERMS

accessibility: making an application accessible to as many people as possible, including those with disabilities

target audience: the typical users (audience) that an application is aimed at

You may also need to consider the users' likely level of language and literacy when designing and developing the user interface, and issues such as making error messages clear and understandable. This can help users who speak a different first language or younger users.

Whatever type of system is being developed, it should be possible to define the characteristics of the target users. This definition should cover a range of characteristics, including things such as age, IT skills, language and literacy.

Aspects of the user experience and the needs of different users are covered in more detail in the next section.

It is important to build **accessibility** in when developing a database system. This will involve considering how you can create an interface that works effectively for all people, including those with disabilities. A few examples of how you can improve accessibility are by:

- using colours with sufficient contrast for text and backgrounds in the interface to make them clearer for users with visual impairments
- providing keyboard shortcuts to select menu options so users who can't use a mouse can still navigate menus
- using input forms with clearly defined borders around text fields to make them easy to see. All input boxes on forms should be clearly labelled

If you search online for WCAG or Web Contents Accessibility Guidelines you will find lots of information about how websites can be made fully accessible.

- providing clear and consistent navigation options, such as a back button to take the user from a submenu to a main menu.

Evaluating the appropriateness of the database design for specific users

It may be hard for the developer to assess how appropriate the database design is for the target users, so an effective approach will be to create a prototype or mock-up of the database and demonstrate it to some users from the target group. Obtaining feedback from this target user group should then enable you to assess how appropriate the design is and what improvements might be needed.

SKILLS

ANALYSIS, CRITICAL THINKING

ACTIVITY 2

A college is considering creating a database for their library. This would record all the books, magazines, journals and newspapers they have in the library. It would also list all the readers (students) who are registered and can take books out on loan.

1. Who is the target audience for the library database? How can the characteristics of the target audience be defined?
2. What methods can be used to collect information about user needs?
3. How can you consider accessibility when developing this database application?
4. If users have low IT skills how should that affect the application you develop?

USER EXPERIENCE

You have already started to consider how the users' needs should be taken into account when designing a database. Now you need to look in more detail at how the user interface of the database needs to be designed so that the **user experience (UX)** of using the application is a positive one. The users should find it easy to use, they should be able to achieve the task they are required to complete quickly and it should be clear how to carry out the task. They should not find the interface confusing or frustrating, and if they are not using the system in the correct way (for example, entering invalid data into a field) then the application should provide clear error messages that allow them to understand what they are doing wrong.

KEY TERMS

graphical user interface (GUI): an interface that uses icons, buttons and menus with a mouse or touch-based pointing method

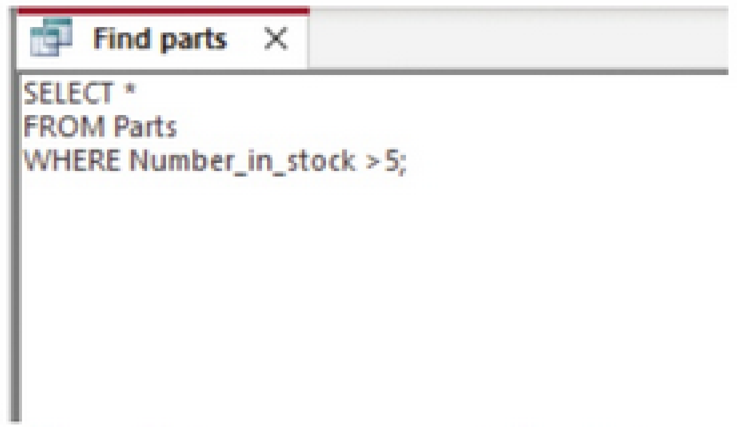
query: a database tool that filters data to match given criteria

user experience (UX): how a user experiences their interaction with a system

DATABASE DESIGN AND USER CHARACTERISTICS

As you have already seen, user characteristics such as age, level of IT skills and language need to be taken into account when designing a database. The design features of the database user interface that need to be considered include the following:

Interface type. There are two main types of user interface, graphical and text based. A **graphical user interface (GUI)** with features such as windows, icons and menus accessed by interaction with a mouse or touch screen is the most commonly used, and will be familiar to most users with any IT skills. A text-based user interface does not have these features, and commands need to be entered using some kind of syntax to input a valid command. This type of interface may suit expert users who are familiar with the command syntax, and they may even find this method faster than having to use a mouse. However, text-based interfaces are not suitable for those with only limited IT knowledge and experience. Figure 17.5 shows an example of a text-based interface being used to create an SQL **query** to select matching records from a table. Queries are explained in more detail in Topic 19.



```
SELECT *
FROM Parts
WHERE Number_in_stock >5;
```

▲ Figure 17.5 Creating a query with a text-based interface

Interface characteristics. It is important that the database application has a professional looking interface that is clear and easy to understand. Not only will this make the application easier to use, but an interface that is messy and unprofessional will cause users to perceive it as a poor-quality product even if it functions correctly. The use of colours, typefaces and text sizes are important for readability. A poor combination of colours may make menus and other text difficult to read; for example, there should always be a contrast between background and text colours, such as a darker colour text on a lighter background.

Complexity. An interface that is complex with many different buttons, options, menus, and so on, may confuse users and will typically make a window look overcrowded. You should consider a logical hierarchy of menus and options, starting with a main menu that has options for the main functions of the application and leading to sub-menus for each main function, if required. If there is a logical progression of tasks through the application then the menus should follow that order, with the task likely to be done first being the first shown on the menu, and so on. Alternatively, if there is no obvious progression through the application, then functions should be listed in order of importance or how regularly they are used, with the most important or most often used at the top of the menu.

Use of language. The language you use in the application needs to match the names and terms that users are familiar with. This includes things like menu names, as well as field and table names. Error messages also need to use appropriate language and clearly explain what the error is and how the user can resolve the issue.

Compatibility. It may be that the database application needs to be used in a variety of different contexts. This is particularly the case with applications that have a web-based front end. Users are likely to have different browsers to access the application, so it is important that the application has been tested with all the commonly used browsers to ensure compatibility.

USER EXPERIENCE AND DATABASE DESIGN

User experience design is also known as UX design. Ideally an application should provide a good experience for the user, with an interface that is easy to use, allows them to complete required tasks efficiently and provides a positive experience.

Interface design. The design of an interface that takes user experience into account is a complex topic that has been researched and written about extensively. This is particularly the case in areas such as web design and mobile app design because organisations want users to have a positive experience with their website and mobile app so that they will return regularly. This is part of the marketing and promotion of the organisation's products. Some key points about designing an interface for users are below.

The design should be user centred: the needs of the user should be the most important consideration when creating the design. Understanding the users (and defining the target audience) and how they might want to use the application is therefore important.

The design should be consistent: it should use layouts, colours, icons and other features in a consistent way.

There should be a hierarchy in the layout of the interface with the most important or most often used buttons or menu items in the most prominent position.

Users should be provided with a way back. Users often click menu items or buttons by mistake and find themselves in the wrong place. Therefore, it is important to provide a 'back' or 'cancel' button to take the user back to where they previously were.

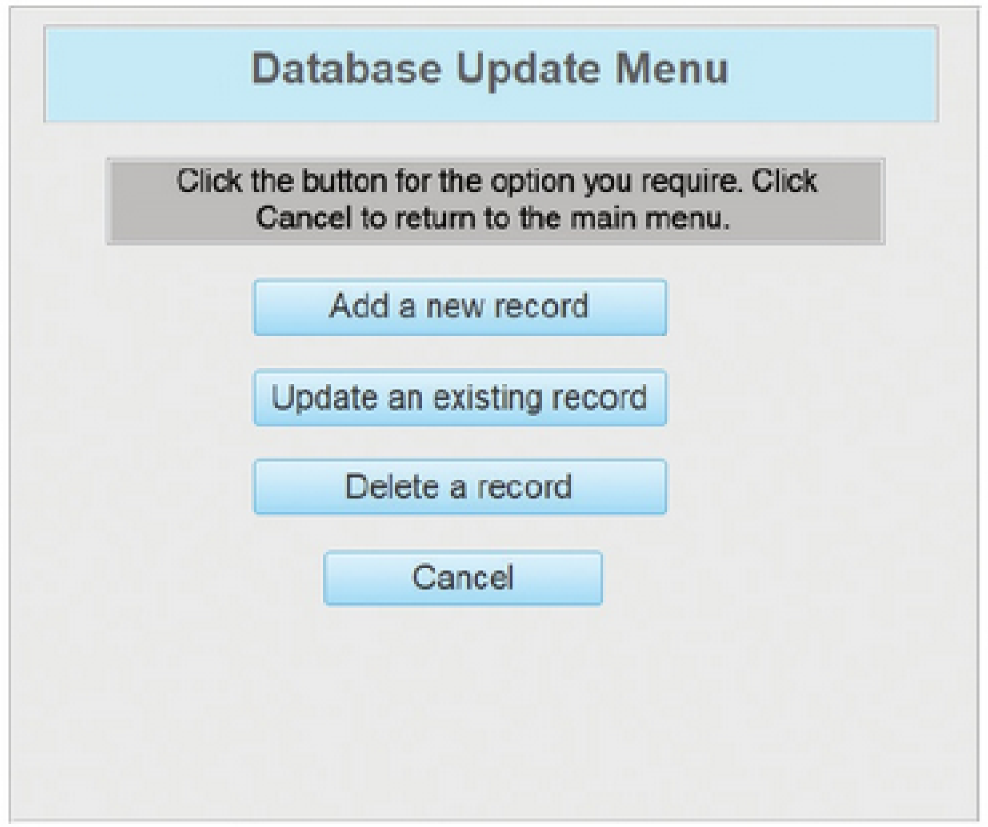
Ease of use. How easy an interface is to use will depend to a degree on the target users. As mentioned earlier, the design needs to reflect their level of IT skill. However, there are some aspects of interface design that will help make it easier to use for anyone.

Clear labelling. Buttons and menus should be clearly labelled. Adding tool tips (small boxes with text in them that pop up when the mouse moves over a button or field) may help users understand how to use a button or what sort of data is expected in an input box.

Predictability. The interface should work in the same way throughout the application. For example, if an icon does something on one menu, it should do the same thing on all the menus.

Keep it simple. Avoid unnecessary complications. It should be easy and quick for users to complete the most common tasks.

Organised content. Organise content in a way that follows the usual workflow that users will follow.



Click the button for the option you require. Click Cancel to return to the main menu.

▲ Figure 17.6 A simple, clear menu

Figure 17.6 shows an example of a simple, clear menu.

One term that is sometimes used to describe an easy-to-use interface is intuitive. This means that it is obvious how to use the interface and it works in a natural way without the need to refer to instructions or help pages.

It can be difficult for a developer to understand what real users will find easy to use, so one approach is to develop a prototype interface and show it to a user group and ask for their feedback. An example of a poorly designed user interface that is unlikely to result in a good user experience is shown in Activity 4, Figure 17.8.

Appropriateness of features. The features of the database application need to be appropriate. The required features are something that should have been identified when researching user needs. Missing or unnecessary features will result in a poor user experience. For example, providing tool tips, help buttons and detailed on-screen instructions for those who have little or no IT knowledge.

Efficiency. This is about how efficiently users can complete tasks. Do they need to work through too many menus to achieve a task that they do often? The most commonly used tasks should be the easiest to find.

User help. Help should be available using a button or icon that provides additional explanation on how to use a particular feature. Tool tips can also be used for this purpose. Help screens, like error messages, must be written in a way that will be understandable for the target user group.

Compatibility. The need for a web-based interface to be compatible with different browsers has already been mentioned. Other relevant aspects of compatibility include the ability of the interface to work effectively on different screen sizes and orientations. This is important if the interface is likely to be used on different devices such as desktop or laptop computers, tablets and mobile devices.

Stability. An application that is unstable, freezes, crashes or produces unexpected results will frustrate users. Any application should be thoroughly tested before being released to users as any issues will quickly become evident once it is used in the real world. Users will quickly lose confidence in an unstable application and it will not be easy to get it back.

Data accuracy. It is important that data integrity and accuracy is maintained. Where calculations are done using the data in the database, they need to be thoroughly tested to make sure the results are correct in every situation.

SKILLS

ANALYSIS, CRITICAL THINKING

ACTIVITY 3

The Post Office in the UK implemented a nationwide electronic point of sale system called 'Horizon', which was supplied by the IT software company Fujitsu.

The system had many coding and data accuracy issues.

1. Do some research into the impact these problems had on the post office workers who were using it.
2. What can you find out about the 'Dalmellington' bug? What do you think could have been done to avoid this problem?
3. What can you find out about the 'Callendar Square' bug? How could this have been avoided when the database was designed?

Understanding user characteristics

Table 17.2 summarises how a suitable user experience might be provided for users with different characteristics.

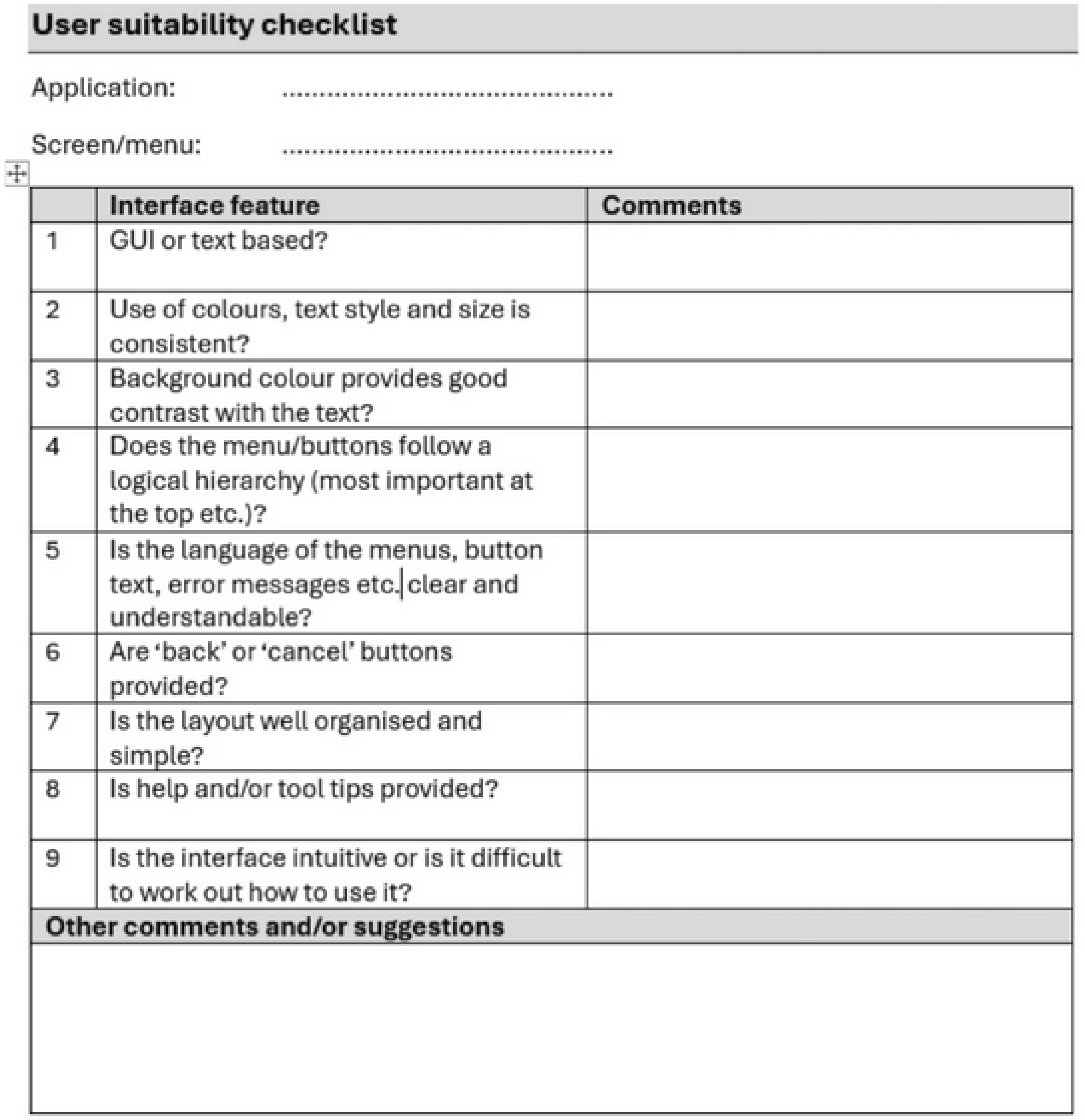
TABLE 17.2: USER CHARACTERISTICS AND INTERFACE FEATURES

User characteristics	Features the database application may need in order to provide a suitable user experience
Very little IT knowledge	Simple interface. Use of tool tips, help buttons, detailed on-screen instructions. Application that guides the user through each step. Error messages should be clear and explain how to resolve the error.
Some IT knowledge	More complex interface, limited on-screen instructions, basic help provided.
Excellent IT knowledge	Maybe a text-based interface. Complex features, no on-screen instructions. Application that allows the user to achieve tasks quickly with minimum support provided.
Limited language skills	Use of simple language, images and icons.
Good language skills	Use of more complex language including technical terms.
Visual impairments	Ensure good contrast between text and background colours. Use larger text sizes, for example 14 point rather than 10 or 12 point. Good text layout and spacing.

Evaluating the effect a database will have on the user experience

Evaluating the features of the database in terms of user experience could be done using a checklist of the requirements for a good user experience described above, and then reviewing the whole database application to see if all the requirements have been met or not. The evaluation would also need to take into account the user characteristics. If the users are specific and known then their characteristics can be used to define the requirements for the evaluation. If the users are not specific, then a user profile based on the target audience for the application can be created.

Figure 17.7 shows an example of a checklist.



User suitability checklist

Application: ....

Screen/menu: ....

	Interface feature	Comments
1	GUI or text based?	
2	Use of colours, text style and size is consistent?	
3	Background colour provides good contrast with the text?	
4	Does the menu/buttons follow a logical hierarchy (most important at the top etc.)?	
5	Is the language of the menus, button text, error messages etc. clear and understandable?	
6	Are 'back' or 'cancel' buttons provided?	
7	Is the layout well organised and simple?	
8	Is help and/or tool tips provided?	
9	Is the interface intuitive or is it difficult to work out how to use it?	

Other comments and/or suggestions

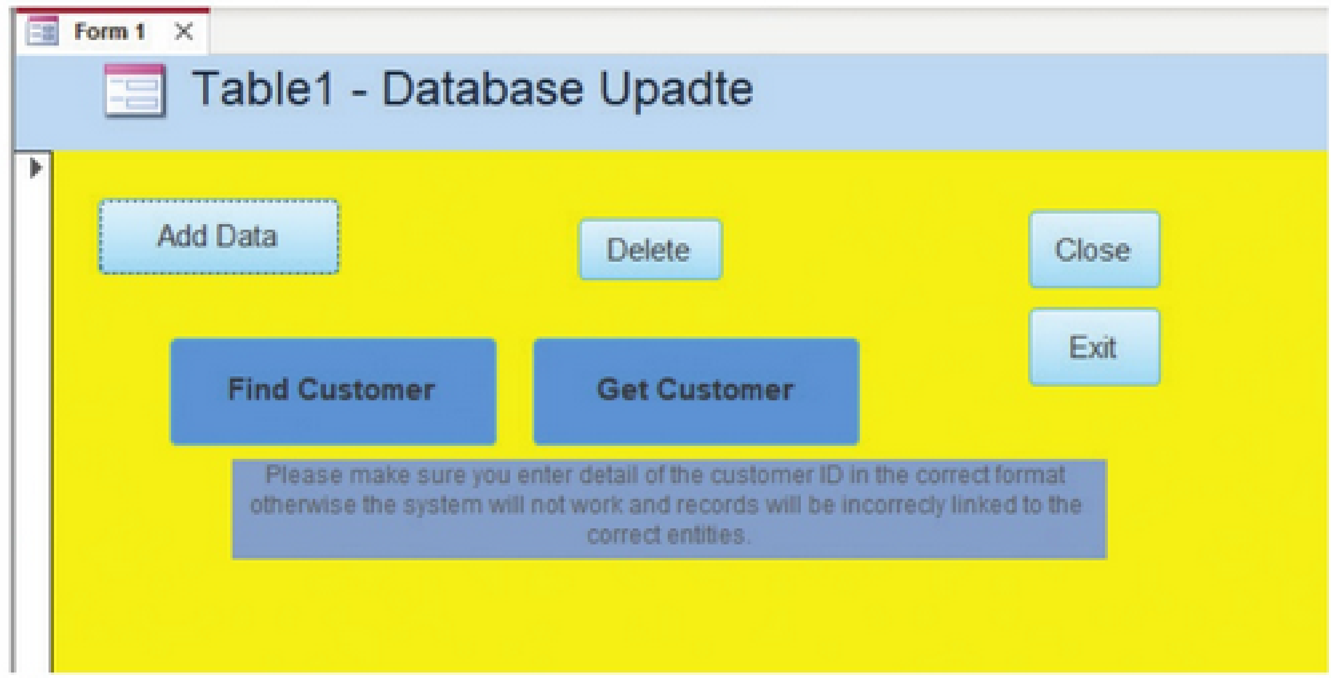
▶ Figure 17.7 Example checklist for evaluating a database application

SKILLS

CRITICAL THINKING

ACTIVITY 4

1. Review the example user interface to a database shown in Figure 17.8. List as many issues as you can identify with this interface.



Please make sure you enter detail of the customer ID in the correct format otherwise the system will not work and records will be incorrecly linked to the correct entities.

▲ Figure 17.8 A poorly designed user interface

2. Find an online database application such as a website that allows you to search for and purchase products or book tickets (flights, trains etc.) and, using the checklist shown in Figure 17.7 (or one of your own making), evaluate the application interface. Who do you think is the target audience for the application?

CHECKPOINT

1. Explain what UX design is.
2. Give three reasons why a database might be used in preference to a spreadsheet.
3. List three business applications that use database technology.
4. Why is avoiding data duplication important?
5. List two aspects of a database application you could consider when evaluating it, apart from user experience.
6. What is meant by the term intuitive?
7. What kind of user is a text-based interface likely to be most suitable for?
8. What kind of interface features would suit users with limited literacy skills?

THINKING LIKE AN IT SPECIALIST

When developing an IT application, understanding the user requirements can be quite difficult, but it is very important. Unless the user requirements are understood and the application designed and developed to fully meet them then the application cannot really be considered successful. Why is it that user requirements can be hard to understand? What can be done to help understand user requirements?

EXAM TIP

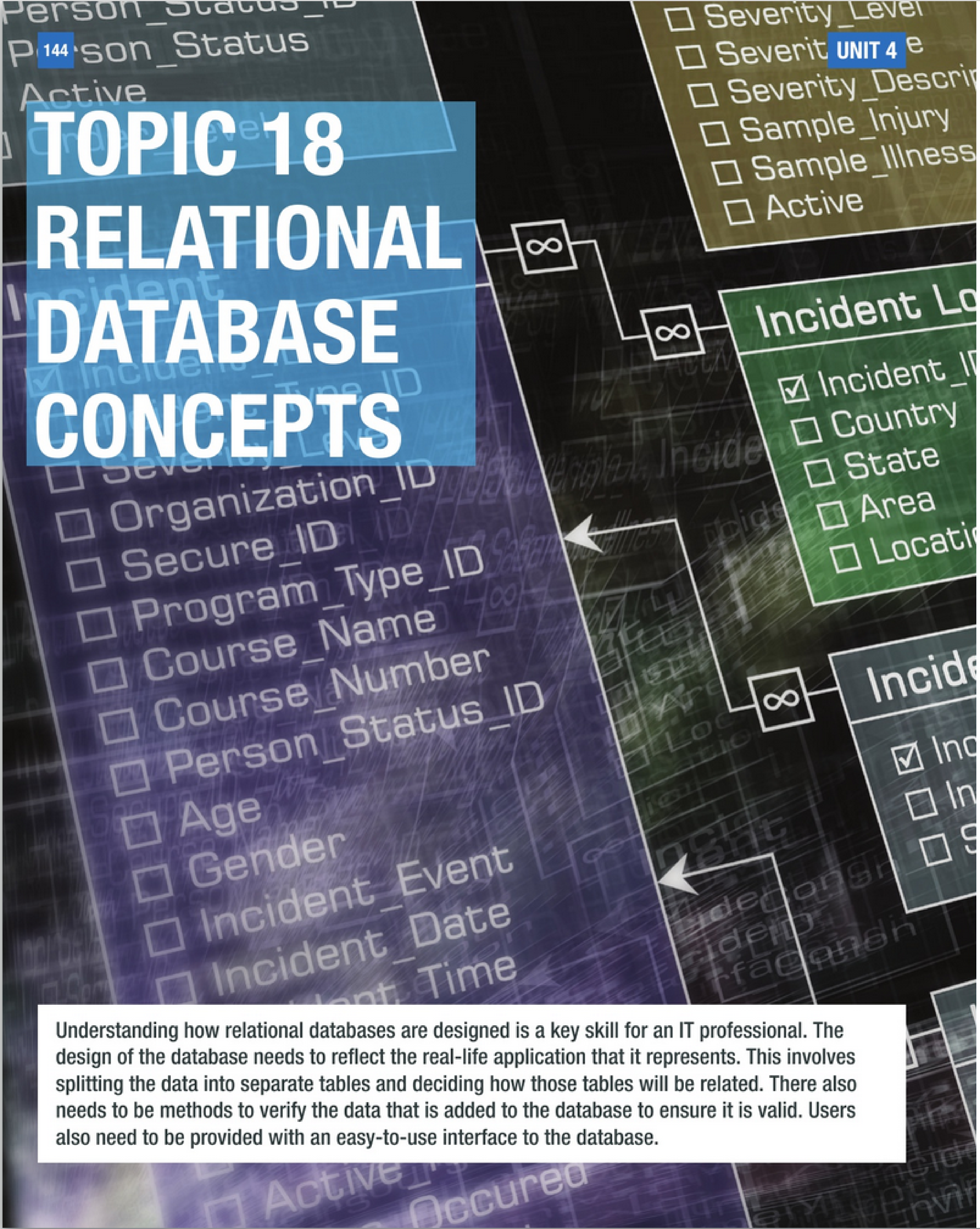
The exam for Unit 4 is a practical one where you will be asked to improve and extend an existing database, adding various features to it. You are likely to be asked to create a form and/or a report, and possibly a dashboard (to provide the user with a menu to access the features you have created) so that you can apply what you have learned about making a user interface professional and user-friendly.

EXAM PRACTICE

1. Give one method that can be used to collect information about user requirements for a database application. [1 mark]
2. Databases can be used to record transactions.

   Describe how databases record transactions. [3 marks]
3. A database application should match the needs of users.

   Describe the features of a database application that would provide a positive user experience to users who have limited IT knowledge. [3 marks]


Person_Status_ID
Person_Status
Active
Severity_Level
Severity_Name
Severity_Description
Sample_Injury
Sample_Illness
Active
Incident
Organization_ID
Secure_ID
Program_Type_ID
Course_Name
Course_Number
Person_Status_ID
Age
Gender
Incident_Event
Incident_Date
Incident_Time
Active
Occured
Incident Lo
Incident_ID
Country
State
Area
Locati
Incide

TOPIC 18 RELATIONAL DATABASE CONCEPTS

Understanding how relational databases are designed is a key skill for an IT professional. The design of the database needs to reflect the real-life application that it represents. This involves splitting the data into separate tables and deciding how those tables will be related. There also needs to be methods to verify the data that is added to the database to ensure it is valid. Users also need to be provided with an easy-to-use interface to the database.

TOPIC 18 RELATIONAL DATABASE CONCEPTS

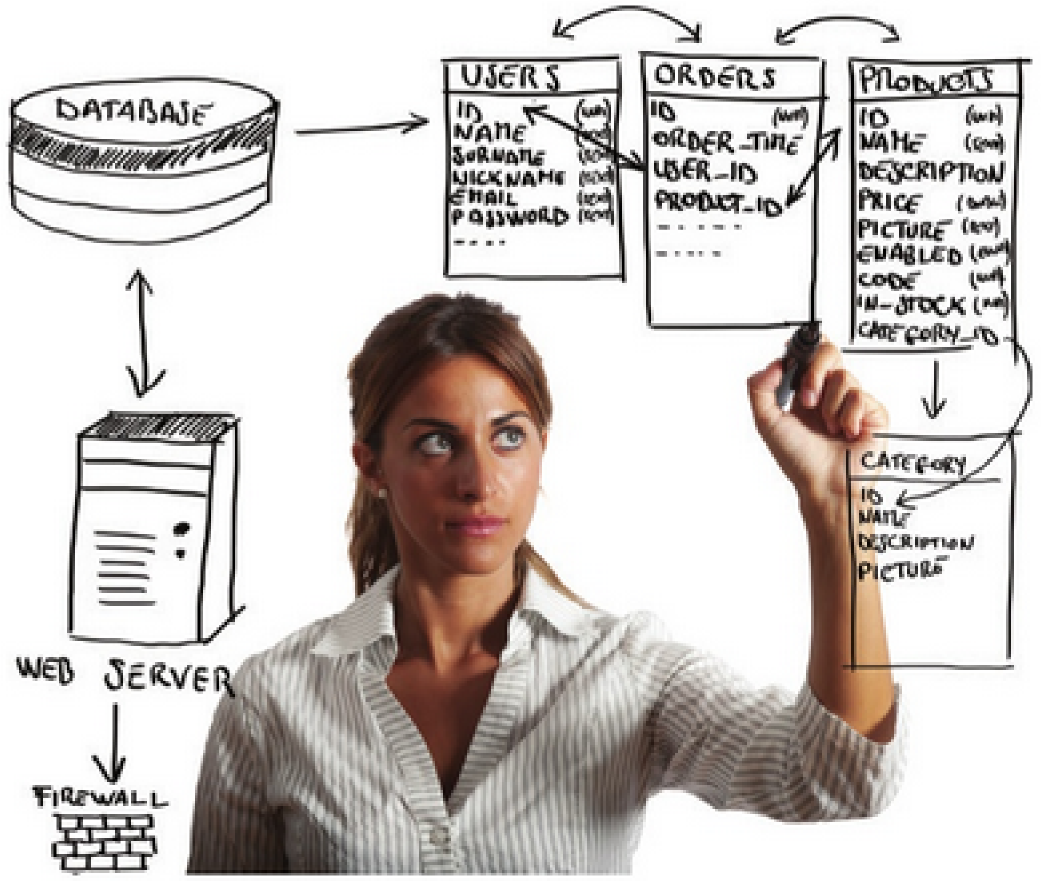
LEARNING OBJECTIVES

By the end of this chapter you should be able to understand:

- how to design the structure of a relational database
- how to use appropriate data types, format them and evaluate a data structure
- how to build database tables
- how to create forms to allow users to input data
- how to make input forms user friendly
- relational data structures including why they are necessary and how to implement data relationships
- how to validate data to ensure accuracy and why this is necessary
- how to improve the quality of data entry and evaluate techniques to do so.

GETTING STARTED

In Topic 17 you were introduced to the idea of database applications and how users interact with them. Think about why users might want to interact with data. What type of tasks might they need to complete? What kind of questions might a user want answered by a database? How might that information about the data be provided?


DATABASE
WEB SERVER
FIREWALL
USERS
ID
NAME
SURNAME
NICKNAME
EMAIL
PASSWORD
ORDERS
ID
ORDER_TIME
USER_ID
PRODUCT_ID
PRODUCTS
ID
NAME
DESCRIPTION
PRICE
PICTURE
ENABLED
CODE
IN_STOCK
CATEGORY_ID
CATEGORY
ID
NAME
DESCRIPTION
PICTURE

▶ If you are going to build a database you need to plan it before you start.

STRUCTURING DATA

CONSTRUCTING AND AMENDING RELATIONAL DATABASES

As you saw in the previous topic, the main structure in a relational database is called a table. A table holds a collection of data and consists of rows, called records and columns, called fields. The fields in a database represent real-world things like customers, orders, prices, weights, dog breeds, flights, etc. An example of the table of fields and records is shown in Figure 18.1. Fields should be atomic, meaning that it should not be possible to split the field further. For example, if you have a field for a person's name, it could store data such as 'Ms Amita Patel'. This is not atomic because it can be further split into their title ('Ms'), their first name and their family name. There should therefore be at least three fields for a person's name.

Fields need a field name, and you should follow some kind of naming convention when selecting names for fields. You should choose a name that is meaningful and describes what the field contains. While some database software will allow spaces in names, you should avoid this as it may cause problems if the data needs to be used in a web page. Names like 'Customername' are fine. There are some reserved words that you can't use for field names (like Date or Number). The same applies to tables.

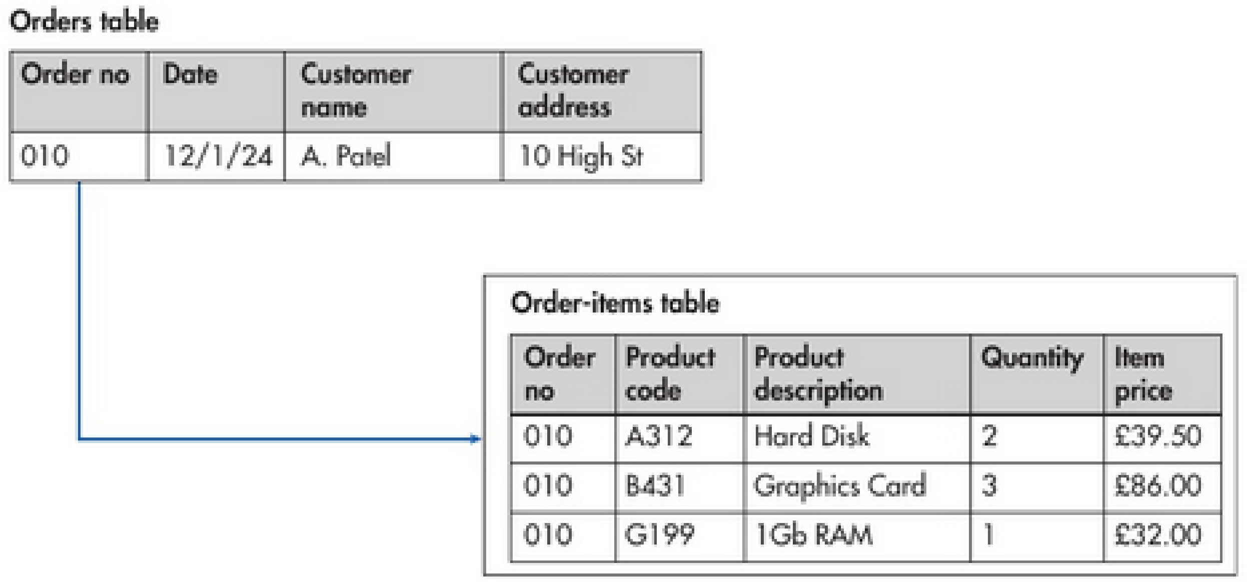
When a relational database is developed, one of the first tasks that needs to be completed is how to divide all the data that needs to be stored into appropriate tables. An important principle of relational databases is that duplication of data should be minimised. A table is a collection of data, but each record in the table needs to describe just one item in the collection. Imagine a database recording transactions such as online sales. While some sales might just involve purchasing one item, many sales would involve purchasing multiple items. Therefore an 'Orders' table could hold information such as the order number, the date of the order and the customer who placed the order. Table 18.1 below shows some of the data that needs to be recorded when an order is placed. In this example the customer has added three items to the same order.

TABLE 18.1: DATA RECORDED FOR AN ORDER

Order_ no	Date	Customer_ name	Customer_ address	Product_ code	Product_ description	Quantity	Item_ price
010	12/1/24	A. Patel	10 High St	A312	Hard Disk	2	£39.50
010	12/1/24	A. Patel	10 High St	B431	Graphics Card	3	£86.00
010	12/1/24	A. Patel	10 High St	G199	1Gb RAM	1	£32.00

As you can see, some data (the details of the customer) is duplicated. Duplicate data can lead to inconsistencies in the data, as explained in the previous topic.

Therefore, because there might be more than one item ordered, the details of what has been ordered would need to go into a separate table, perhaps called 'Order-items' (see Figure 18.1). The link between the 'Orders table' and the 'Order-items table' is made by the 'Order No' field.



Orders table

Order no	Date	Customer name	Customer address
010	12/1/24	A. Patel	10 High St

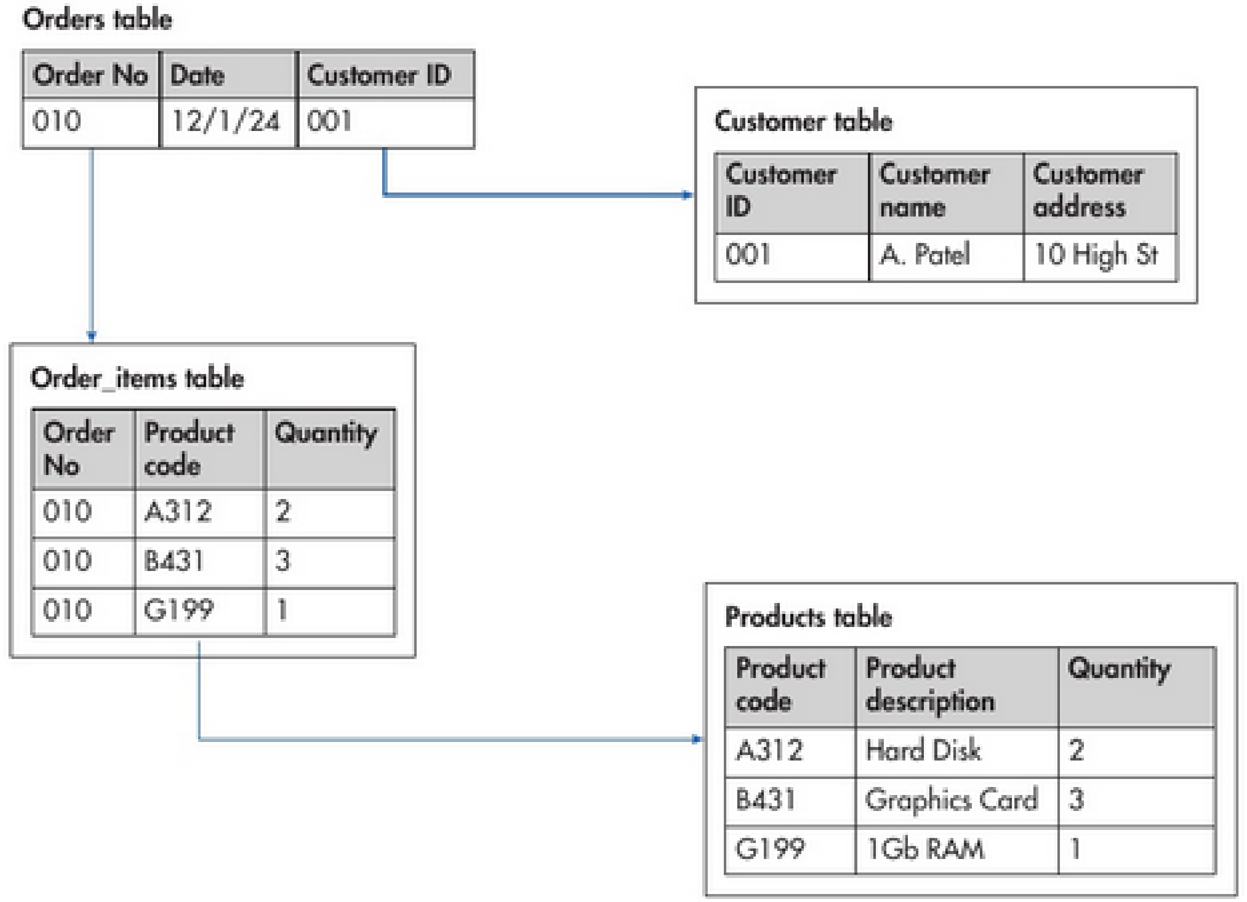
Order-items table

Order no	Product code	Product description	Quantity	Item price
010	A312	Hard Disk	2	£39.50
010	B431	Graphics Card	3	£86.00
010	G199	1Gb RAM	1	£32.00

▶ Figure 18.1 Order data split into Orders table and Order-items table

The Orders table also needs to record the customer who placed the order. However, it is likely that a customer would place many orders over time, so putting all their details in the Orders table would create duplication because their details would be repeated for each order that they place. Therefore, the customer details need to be placed in a separate table, perhaps called the 'Customer table'. So you have now divided the data into three tables. However, there is still potential for duplication.

The Order-items table needs to contain details of the products ordered. However, products are likely to be ordered many times, in different orders. Therefore, the details of products (their description, price, etc.) need to go in a different table, so there can be just one record per product. That table could be called the 'Products table'. The database now has four tables as shown in Figure 18.2.



Orders table

Order No	Date	Customer ID
010	12/1/24	001

Customer table

Customer ID	Customer name	Customer address
001	A. Patel	10 High St

Order_items table

Order No	Product code	Quantity
010	A312	2
010	B431	3
010	G199	1

Products table

Product code	Product description	Quantity
A312	Hard Disk	2
B431	Graphics Card	3
G199	1Gb RAM	1

▶ Figure 18.2 The four related tables of the Orders database

Another important principle of relational databases is that each record in a table should be unique. To this end, each record is usually given a unique identifier, which is called a key. In the Customer table you could use the customer's name as the key, but as there might be several people with the same name, a better idea is to use a unique identifier such as a customer number, or 'Customer ID'. When a customer places an order, a record is created in the Orders table, but it needs to be related to a record in the Customer table (the customer who placed the order). That relationship is created by placing the Customer ID in the Orders table. Every order that each customer places is therefore identified by including the Customer ID from the Customer table in the orders placed by that customer. The two tables are related by the Customer ID.

SKILLS

ANALYSIS, PROBLEM SOLVING

ACTIVITY 1

1. Imagine you are developing a database design for a hospital. Below is some of the data that needs to be recorded.

Ward no	Ward name	Ward type	Bed no	Bed occupied	Patient no	Patient name	Patient date of birth
1	Daisy	Surgical	1	Yes	125	Patel	10/1/61
1	Daisy	Surgical	2	No			
1	Daisy	Surgical	3	Yes	184	Vine	1/6/79
1	Daisy	Surgical	4	No			
2	Spring	General	1	Yes	762	Ramerez	19/5/91
2	Spring	General	2	No			

Looking at this data, where does duplication occur? How could this data be split into separate tables to minimise the duplication?

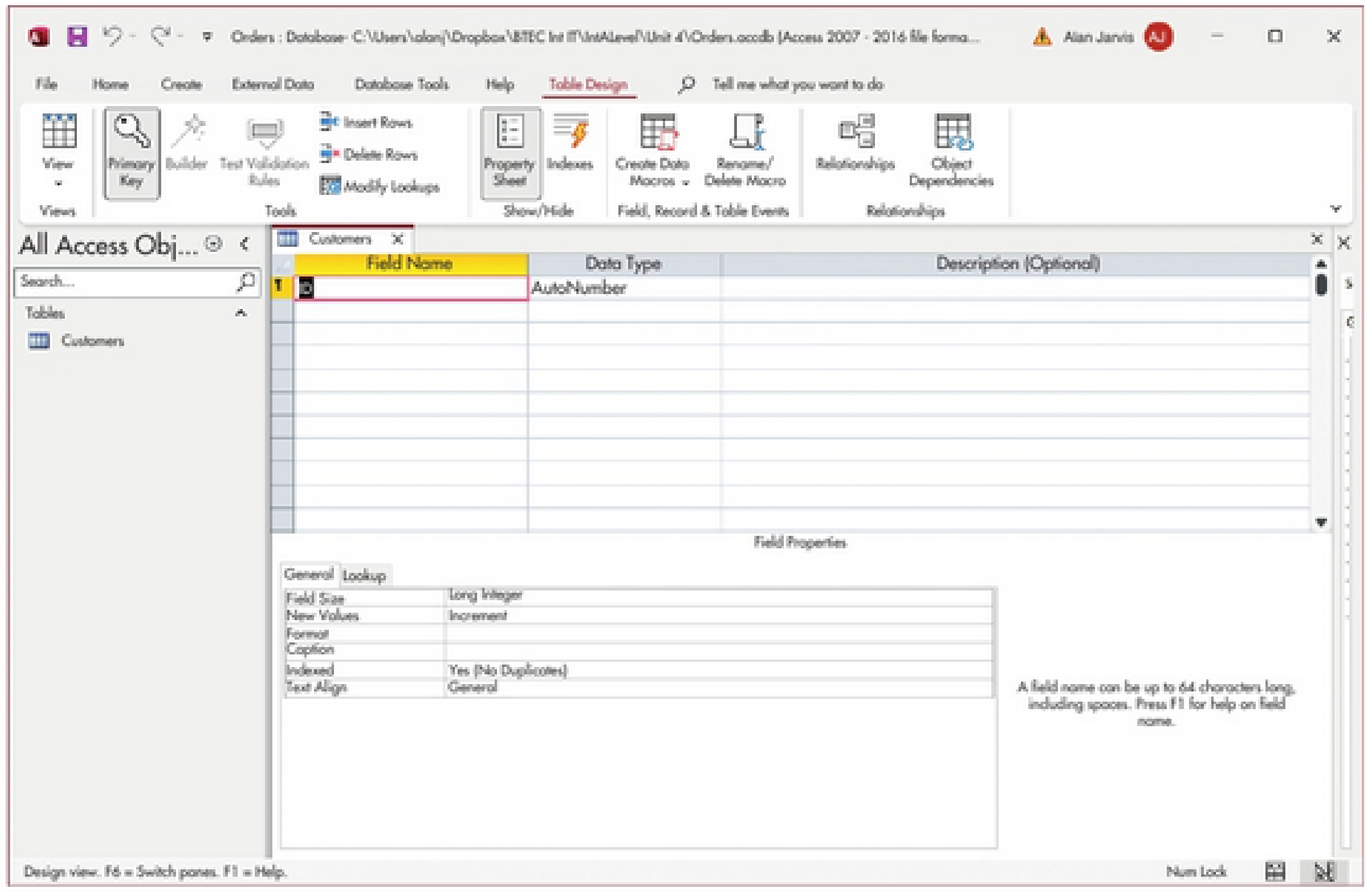
List the names of the tables you think the data can be split into.

Which fields belong in which table?

Draw a diagram like the ones above for this data by hand.

CONSTRUCTING TABLES AND FIELDS

Let's see how you can create tables and fields in Microsoft Access. When Access starts you can choose to create a new blank database; when you do this it creates a table called 'Table1'. To customise this table to meet your requirements, choose View on the top left of the toolbar and select Design View. This will change your view to Design view, as shown in Figure 18.3. You will be asked to save the table, giving it a name. In the example below, the table is called 'Customers'.


Orders : Database- C:\Users\alanj\Dropbox\BTEC Int IT\IntALevel\Unit 4\Orders.accdb (Access 2007 - 2016 file forma...
Alan Jarvis
AJ
File
Home
Create
External Data
Database Tools
Help
Table Design
Tell me what you want to do
View
Primary Key
Builder
Test Validation Rules
Insert Rows
Delete Rows
Modify Lookups
Property Sheet
Indexes
Create Data Macros
Rename/ Delete Macro
Relationships
Object Dependencies
Views
Tools
Show/Hide
Field, Record & Table Events
Relationships
All Access Obj...
Search...
Tables
Customers
Customers
Field Properties
General
Lookup
Design view. F6 = Switch panes. F1 = Help.
Num Lock

Field Name	Data Type	Description (Optional)
ID	AutoNumber	

Field Size	Long Integer
New Values	Increment
Format	
Caption	
Indexed	Yes (No Duplicates)
Text Align	General

A field name can be up to 64 characters long, including spaces. Press F1 for help on field name.

▶ Figure 18.3 Table Design view

KEY TERM

datatype: a field property that sets the sort of data that can be stored in the field

The table will by default have a single field already added called ID. This can be used as a unique identifier for each record in the customer table, so you can change the name to Customer_ID. Access has set this to an AutoNumber field. You will look at **datatypes** in the next section, but an AutoNumber field is automatically numbered by Access, starting at number 1. Access has also made this a primary key field, as shown by the key symbol to the left of the field name. Keys are also covered in a later section. You can add further field names for the fields you want in your table and then save the table by clicking the Save icon in the top left of the Access window. You can always modify the design of your table, although it is best to get the table design complete before you add a lot of data to the table. You can close the Table Design view by clicking the X next to the table name above the list of field names.

Datatypes

As well as giving a field a name, you will also need to set a datatype for the field. As with programming, the datatype sets the sort of data you can store in a field. It is important to use the correct datatype so that things like calculations and queries work as you expect.

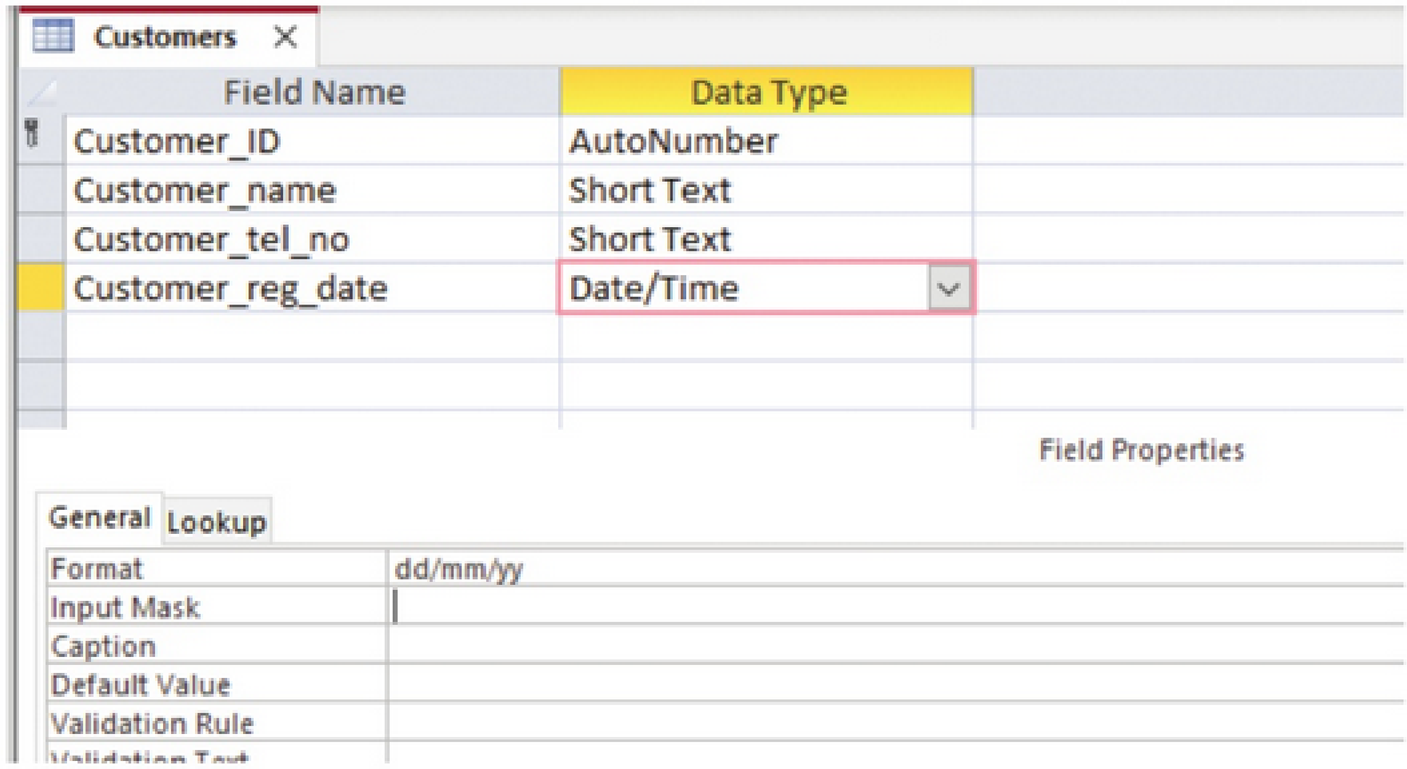
The most commonly used datatypes are:

Text. Used to store data that is text, including numbers and symbols. Short Text is the default datatype in Access and allows a field length of up to 255 characters. If you select a field that has the datatype of short text in Table Design view, then the Properties panel (below the field names) will show the field size (which by default is set to the maximum of 255 characters). In the Data Type column you can click on the drop-down arrow to display a list of available datatypes. An alternative to Short Text is Long Text. Long Text datatype used to be called 'Memo', and it allows very large amounts of text to be stored in a field (up to about 1 gigabyte).

Number. Used to store numbers of various sizes. The default size in Access for fields set to number datatype is called 'long integer'. The different sizes available for numeric fields are:

- byte – stores whole numbers from 0 to 255 only
- integer – stores whole numbers (positive and negative) from −32,768 to 32,767
- long integer – stores whole numbers from around −2,147 million to 2,147 million
- double – used for floating point numbers, up to about 15 digits long
- decimal – used for fixed point numbers where precision is important (such as scientific research results).

Date and time. Used to store dates and times. You can choose a variety of formats to suit your purpose, such as 'short date', which is in the format dd/mm/yyyy (although this will depend on the region setting in Windows; for example, if your computer is set to the US format, the short date format will be mm/dd/yyyy). You can set your own customised format if you wish. For example, if you wanted a short date format of dd/mm/yy you can enter that in the format box on the datatype property window, as shown in Figure 18.4.


Customers
Field Properties
General
Lookup

Field Name	Data Type
Customer_ID	AutoNumber
Customer_name	Short Text
Customer_tel_no	Short Text
Customer_reg_date	Date/Time

Format	dd/mm/yy
Input Mask	
Caption	
Default Value	
Validation Rule	

▶ Figure 18.4 Setting a custom date format

Currency. A special type of number preformatted with a currency symbol and set to two decimal places.

Boolean. This datatype can only be set to one of two values, true or false, although it can be formatted as other binary values such as yes/no or on/off.

Evaluating the choice of datatypes

It is important that the fields in the database can store the required data and that data can be processed in the required ways. It is also important that the data storage requirements are minimised. This might not seem so important with a database that stores just a few hundred records, but it is a consideration when the database might store millions of records. Therefore, the most appropriate datatype for a field needs to be used. For example, in the Order-items table described above there is a field for the quantity of the item ordered. It is unlikely (depending on the application) that customers will order more than 30,000 items at a time, so a datatype of integer rather than long integer is appropriate.

A common mistake is to use a datatype of number for a telephone (or mobile) number, but telephone numbers are not numbers that you need to do maths with, and they commonly start with zero. Setting a numeric datatype will not allow the field to start with a zero, and it will also prevent you adding international dialling codes in the commonly used format of a plus sign followed by two or three digits (such as +971 for Dubai). Telephone numbers should therefore be set to a datatype of text.

When using the text datatype you should set the length to an appropriate value. Access sets the length of a text field to 255 characters, which is too large for many applications. For example, a person's first or family name or the first line of their address should not need to be any longer than 50 characters.

SKILLS
ANALYSIS

ACTIVITY 2

1. Create the tables for the orders database you have been looking at and set appropriate datatypes for them.
2. The Customer table contains a field of 'Address'. Does this follow the rule that fields should contain 'atomic' data? How should this field be modified?
3. In Activity 1 you designed a hospital database. What datatype should be used for the fields in that database?

RELATIONAL DATABASE STRUCTURES

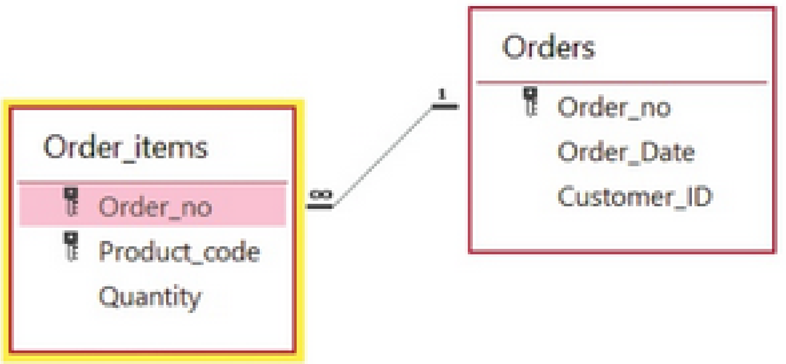
You have already seen how in a relational database, tables can be related or linked to each other. The example used was of a customer ordering goods. One table was used to store data about the order, but the data about the customer was kept in a separate table to prevent adding all the details of the customer to every order record. The tables were linked using the customer ID, which was placed in the records on both tables. This means that the customer details only needed to be recorded once, no matter how many orders the customer placed. One customer record can be related to as many different order records that the customer places. However, in the order table, each record is related to one and only one customer, since the same order cannot be placed by two different customers. This is called a **one-to-many relationship**, and is the most common type of relationship.

Database designers sometime use a diagram called an **entity relationship diagram (ERD)** to represent the relationships between tables (and later you will see that Microsoft Access will draw these for you). ERDs show the tables as a box with the fields listed and a line joining the related tables. The line has an infinity symbol at the 'many' end of the relationship and a '1' at the 'one' end. Figure 18.5 shows an example.

KEY TERMS

entity relationship diagram (ERD): a diagram used by database designers to show the tables in their database and the links between them

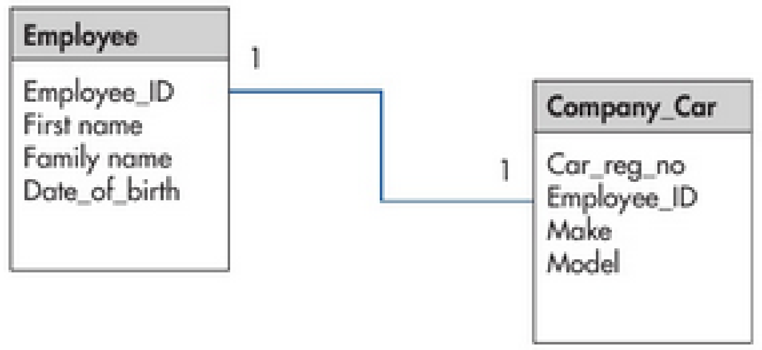
one-to-many relationship: where one record in a table is related to many records in another table



▶ Figure 18.5 An ERD showing the relationship between the Orders and Order_items tables

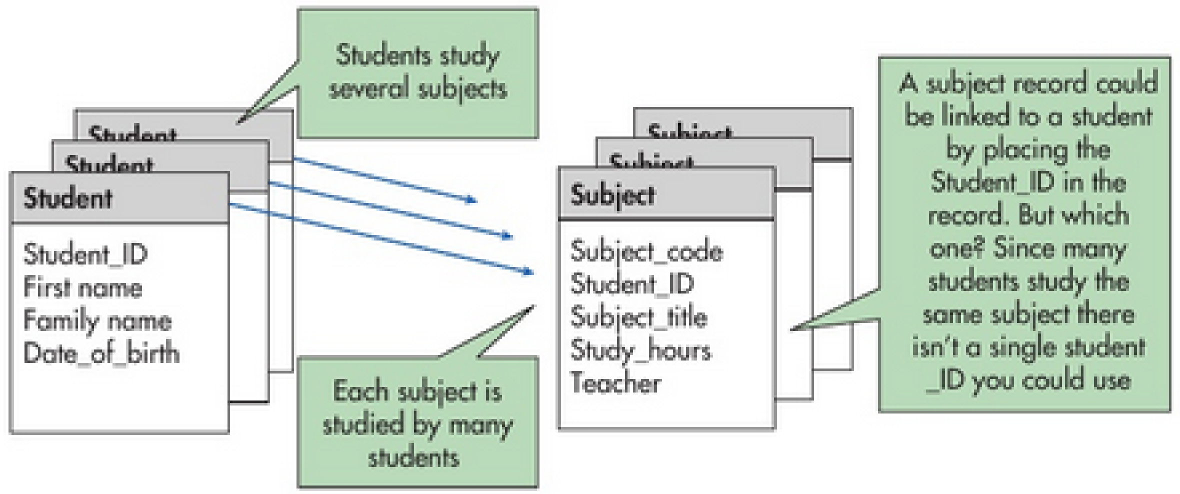
Although a one-to-many relationship is the most common, there are two other types of relationships that can be used.

One-to-one. With this type of relationship, one record in the first table is related to one and only one record in the other table. An example of this might be an employee database. Some employees might have a company car or van. Data about company cars is likely to be kept in a separate table from general employee data, because it is not data about the employee and not every employee will have a company car or van. However, no employee can have more than one company car, so the relationship between the Employee table and the Company Car table is a one-to-one relationship. Every record in the Company Car table is related to one and only one record in the Employee table. Similarly, one and only one record in the Employee table is related to a single record in the Company Car table. Figure 18.6 shows an ERD of this relationship.



▶ Figure 18.6 An ERD showing the relationship between the Employee and Company Car tables

Many-to-many. The final type of relationship is the many-to-many relationship where one record in the first table can be related to many records in the second table, and one record in the second table can also be related to many records in the first table. Imagine a database of college students and subjects that they take. In the Student table, each student record will be related to many subjects (in the Subject table) that each student takes. However, in the Subject table, each subject record will be related to the many different students that take that subject (Figure 18.7).



Students study several subjects

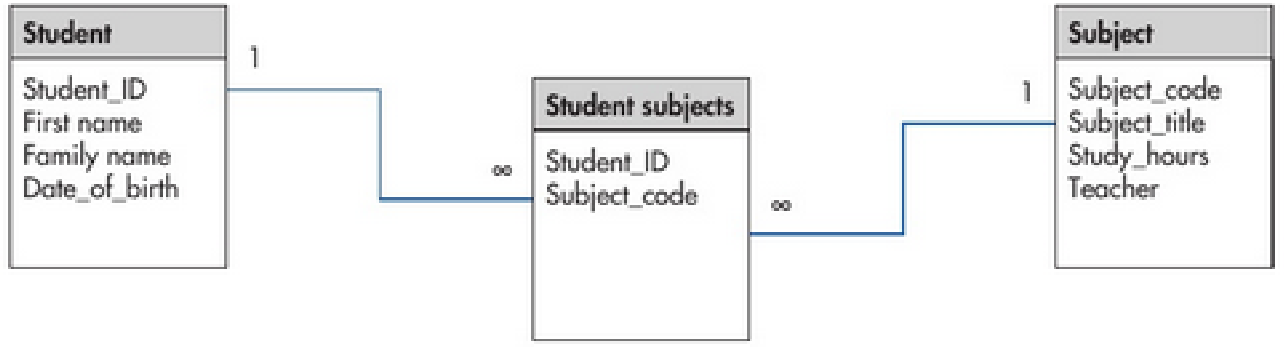
Each subject is studied by many students

A subject record could be linked to a student by placing the Student_ID in the record. But which one? Since many students study the same subject there isn't a single student _ID you could use

▶ Figure 18.7 A many-to-many relationship

Many-to-many relationships cannot be represented directly in a relational database and have to be broken down into three tables with a link or mapping table linking the two main tables.

Let's see how this works with the student and subject example. As already mentioned, linked tables require a unique identifier for each record, so in this example the unique identifier for the Student table is likely to be a student ID, while for the Subject table it is likely to be a subject code. To make the mapping table (which we will call Student-subjects), take the unique identifier attribute from each table and use this to create a link between students and subjects. Where a single student is, for example, taking four subjects, then four records are created in the Student-subjects table, all with the same student ID but with one record for each of the subject codes for the subjects they are taking (Figure 18.8).



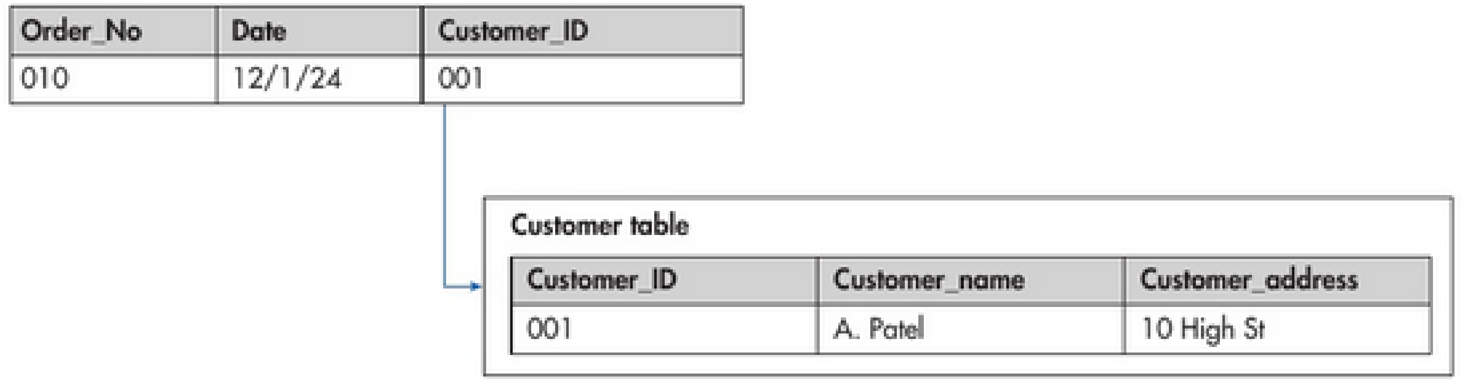
▶ Figure 18.8 An ERD showing the relationship between students and subjects

The Student table has a one-to–many relationship with the Student-subjects table (one student can be related to many mapping table records), and the Subject table has a one-to-many relationship with the Student-subjects table, so one subject record is related to many records in the Student-subjects table.

KEYS

You have already seen the need for unique identifiers such as a customer number (or ID) in the tables of a relational database. These unique identifiers are known as keys. Keys not only uniquely identify records in a database, they can also be used to speed search for records. This is because the database keeps an index of keys so it can find key values without searching through the whole database. The unique identifier for a table is known as the primary key. Each table in a relational database should have a primary key and within the table, each key value must be unique.

When creating a relationship between two tables, the primary key value of one table is inserted into the table that it is linked to. So, using the example discussed earlier of the online ordering system, orders are related to the customer that ordered them by taking the customer ID (the primary key) from the Customer table and putting it into the Orders table against orders placed by that particular customer, as shown in Figure 18.9.



Order_No	Date	Customer_ID
010	12/1/24	001

Customer table

Customer_ID	Customer_name	Customer_address
001	A. Patel	10 High St

▶ Figure 18.9 Orders can be related to each customer using the customer ID.

Of course, these values cannot be used as the primary key value in the Orders table because a customer can place many orders, so it would not be unique in the Orders table. Instead, the Orders table would most likely have a primary key of order number (shown as 010 in the diagram). The customer number in the Orders table is known as a foreign key because it is a key from another table.

KEY TERM

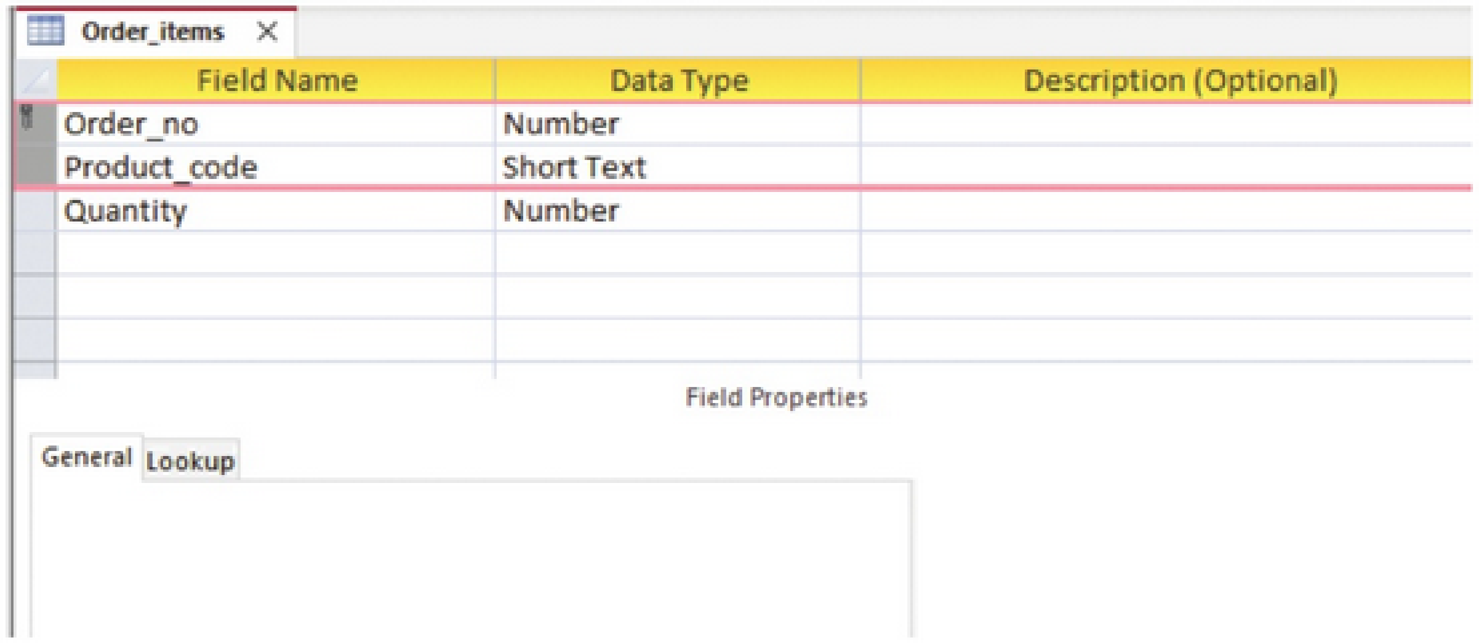
composite key: a key that is made up of two or more fields

In some situations it may be necessary to create a primary key for a table out of more than one field. Take the Order_items table you looked at earlier. What should the primary key be for that table? You could create a field and give it a unique number, calling it Order_items_ID or similar. Another option would be to combine the existing fields of Order number and Product code. By themselves, neither of these two fields would be unique in the Order_items table, but a single order would not have the product code occurring more than once, so together they make a unique key. Combining two fields like this makes a key is known as a **composite key**.

You have already seen that Access makes the first field in a new table the primary key for the table. It also makes the field an AutoNumber field, but you can of course change this if you want to. Remember that the primary key should contain unique values, so if you are not going to use an AutoNumber field then you need to consider how the primary key value will be generated.

However, in this example you cannot use AutoNumber for the Order_number field as the order number you insert in here for a new record is the foreign key of the order to which this item belongs. You therefore need to change the Datatype to Number before you create the composite key.

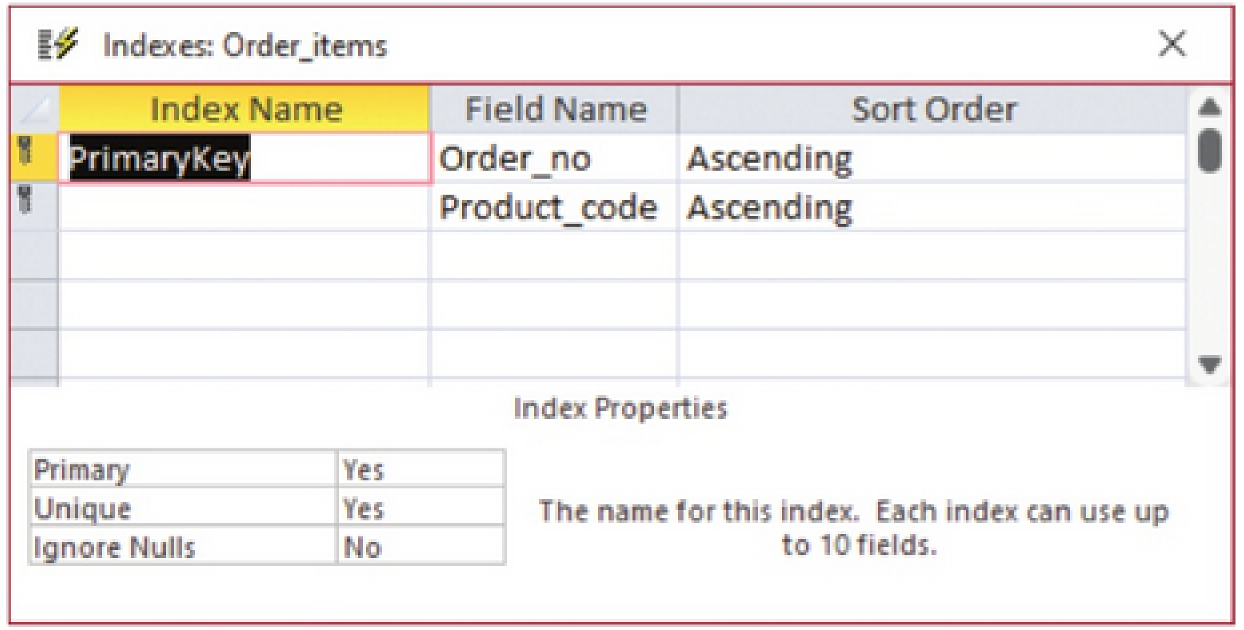
To create a composite key with Access you need to be in Table Design view, then drag your mouse down the grey border to the left of the field names so that both the first and second field are selected, as shown in Figure 18.10.



Field Name	Data Type	Description (Optional)
Order_no	Number	
Product_code	Short Text	
Quantity	Number	

▶ Figure 18.10 Selecting the first two fields in the Table Design view

Now click the primary key button in the toolbar and the key symbol will appear next to both of the fields. You have now set these two fields as a composite key. You can check this is the case by clicking the Indexes icon on the toolbar, which should now look like Figure 18.11.


Indexes: Order_items
Index Properties

Index Name	Field Name	Sort Order
PrimaryKey	Order_no	Ascending
	Product_code	Ascending

Primary	Yes
Unique	Yes
Ignore Nulls	No

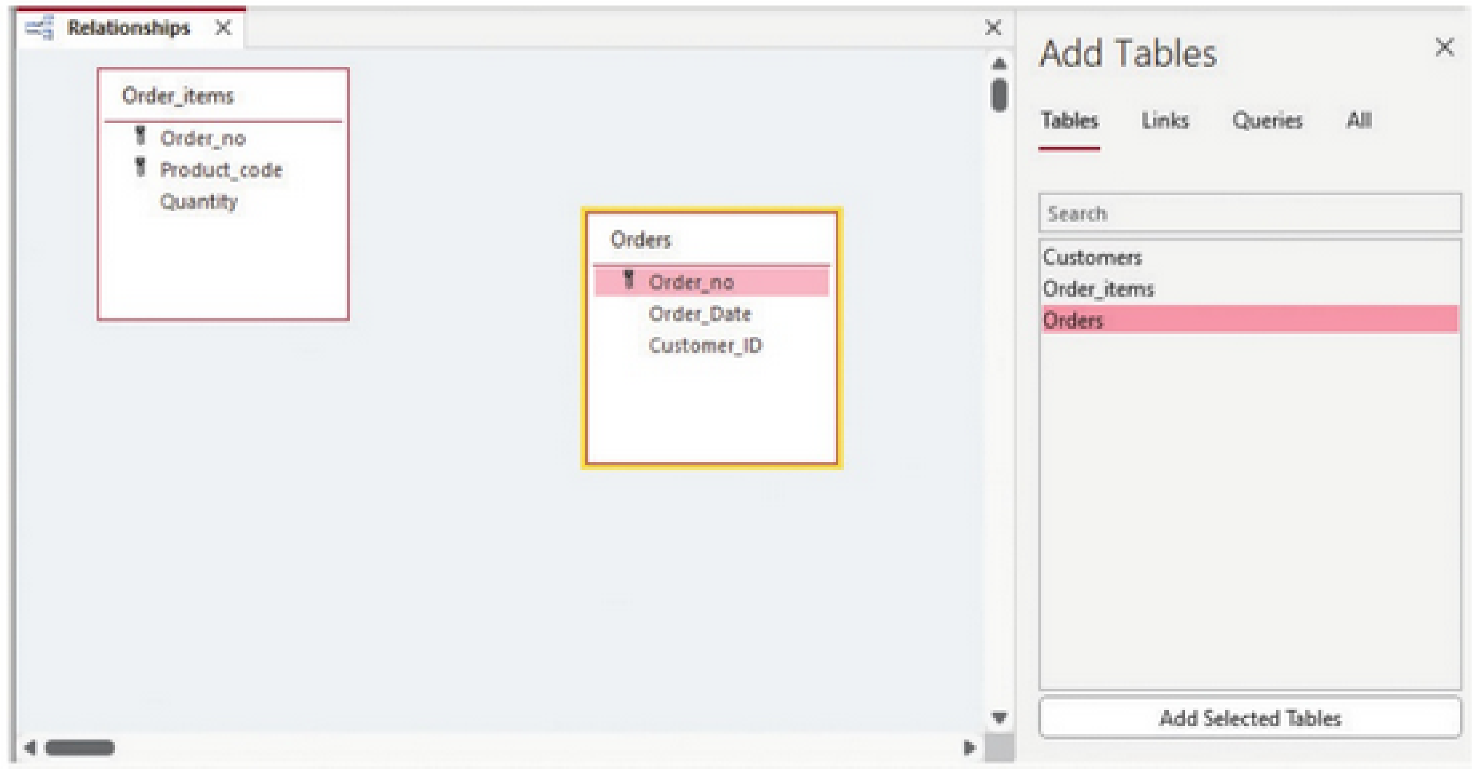
The name for this index. Each index can use up to 10 fields.

▶ Figure 18.11 The Indexes dialogue box, showing a composite key

REFERENTIAL INTEGRITY

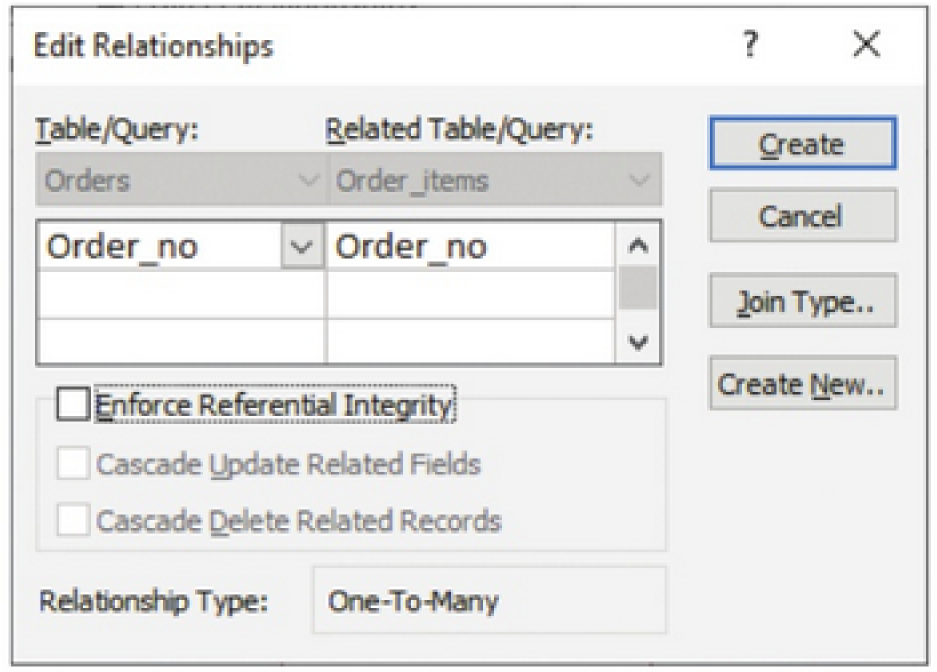
You have looked at database relationships, including one-to-many and one-to-one relationships. These are the basis of relational databases, and in many situations the relationships are essential to the integrity of the database. Looking again at the Orders database, we stated that there is a relationship between the order and the customer table, linked by the customer_ID. This is a mandatory relationship in that an order MUST be linked to a customer. You cannot have orders unrelated to a customer. To create the relationships in Access you need to first create the tables, add the fields and set up the primary keys. Make sure that all the tables are saved and closed before creating the relationships.

Then go to the Database Tools menu and click the Relationships icon, which will open the Relationships window. If you cannot see the Add table panel, click the Add Table icon, click on the tables you want to add and then click the Add Selected Tables button. Figure 18.12 shows the Orders and the Order_items tables added to the Relationships window.


Relationships
Order_items
Order_no
Product_code
Quantity
Orders
Order_no
Order_Date
Customer_ID
Add Tables
Tables
Links
Queries
All
Search
Customers
Order_items
Orders
Add Selected Tables

▶ Figure 18.12 Tables added to the Relationships window

Remember that one order can have many items (products) on it, so the 'one' end of the relationship is the Orders table and the 'many' end is the Order_items table. Click and drag on the Order_no field in the Orders table (the 'one' end), to the Order_no field in the Order_items table (the 'many' end). This will display the Edit Relationships dialogue, which is shown in Figure 18.13.

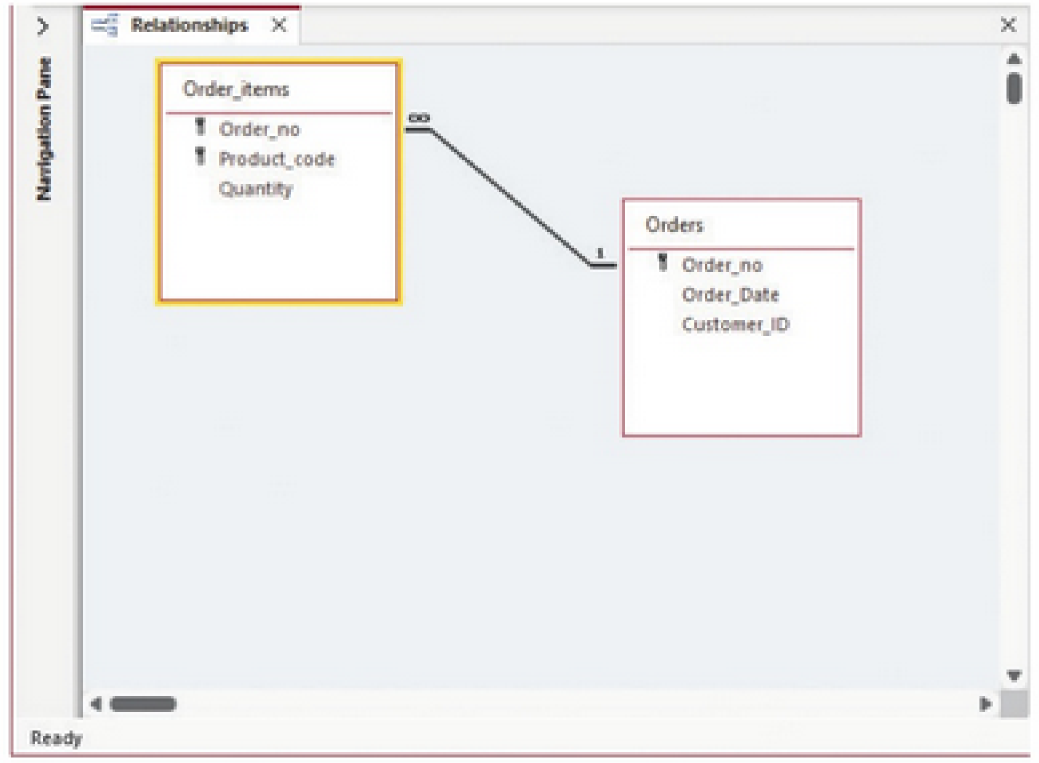

Edit Relationships
?
Create
Cancel
Join Type..
Create New..

Table/Query: Orders
Related Table/Query: Order_items
Order_no | Order_no
☐ Enforce Referential Integrity
☐ Cascade Update Related Fields
☐ Cascade Delete Related Records
Relationship Type: One-To-Many

▶ Figure 18.13 The Edit Relationships dialogue

Check that the correct tables are shown in the correct order (with the Orders table on the left), and then click the check box labelled Enforce Referential Integrity. What this does is prevents a record being created in the Order_items table with an order number that does not exist in the Orders table.

The Relationships windows should now look like Figure 18.14, with an ERD drawn and the 'one' and 'many' symbols shown on the relationship line between the tables.


Navigation Pane
Relationships
Order_items
Order_no
Product_code
Quantity
∞
1
Orders
Order_no
Order_Date
Customer_ID
Ready

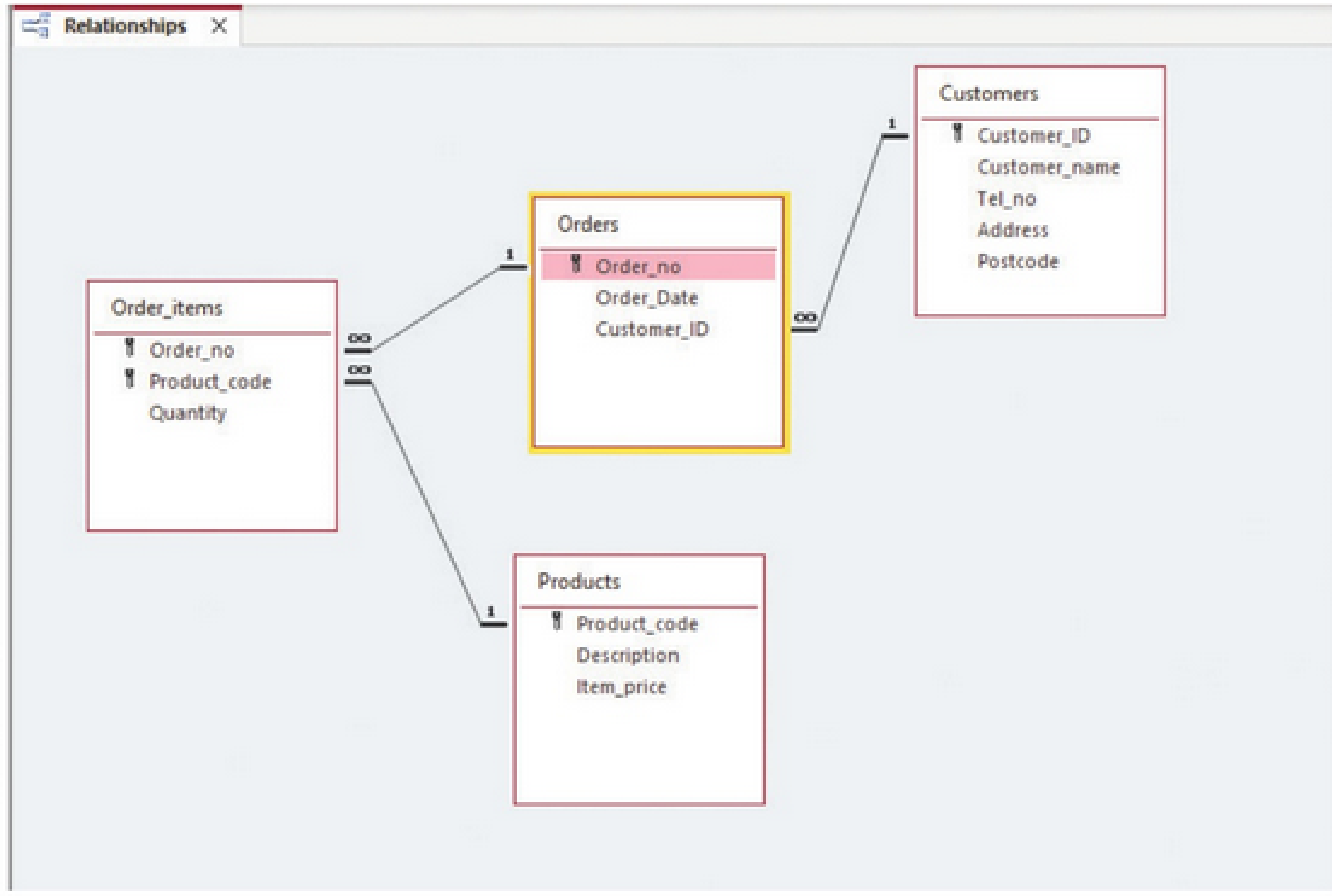
▲ Figure 18.14 The one-to-many relationship between Orders and Order_items

Having created this relationship and selected the **Referential Integrity** option, you will find that if you try to create a record in the Order_items table with an Order_no that

KEY TERM

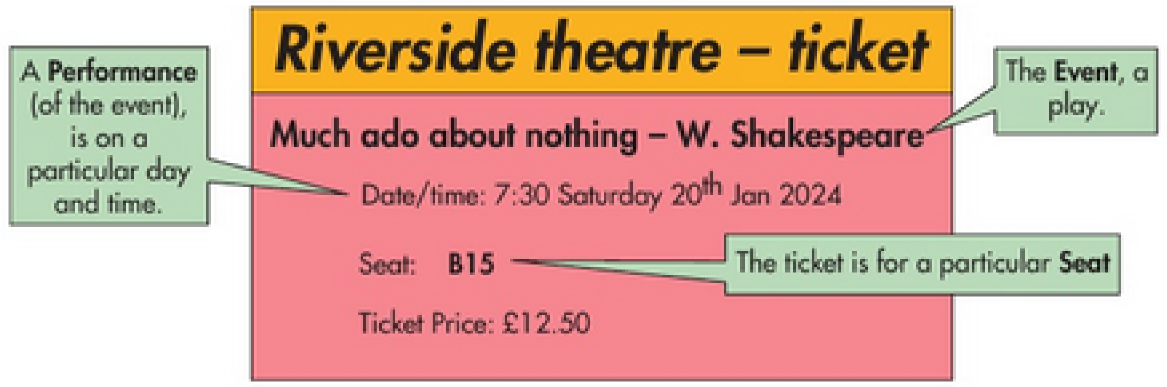
referential integrity: a setting in Microsoft Access that prevents you creating a record in a linked table at the 'many' end using a foreign key that does not exist in the table at the 'one' end does not exist in the Orders table, you will get an error message telling you that a related record is required.

To complete the process of creating the relationships for this database, you would also need to add a relationship between the Order_items table and the Products table, as well as between the Customers table and the Orders table. The completed relationship window is shown in Figure 18.15.


Relationships
Customers
Customer_ID
Customer_name
Tel_no
Address
Postcode
Orders
Order_no
Order_Date
Customer_ID
Order_items
Order_no
Product_code
Quantity
Products
Product_code
Description
Item_price
1
1
1
∞
∞
∞

▶ Figure 18.15 Evaluating the effectiveness and appropriateness of a relational database design

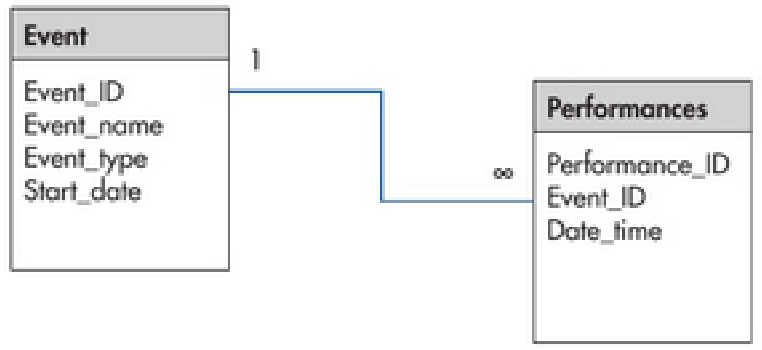
Sometimes, when designing a database, an IT professional may carry out document analysis of an existing paper-based system (or a spreadsheet) that is currently used to record data for an application. An example might be a database needed for booking tickets in a small theatre. Figure 18.16 shows an example of a theatre ticket.


Riverside theatre – ticket
Much ado about nothing – W. Shakespeare
Date/time: 7:30 Saturday 20th Jan 2024
Seat: B15
Ticket Price: £12.50
A Performance (of the event), is on a particular day and time.
The Event, a play.
The ticket is for a particular Seat

▶ Figure 18.16 Riverside theatre ticket

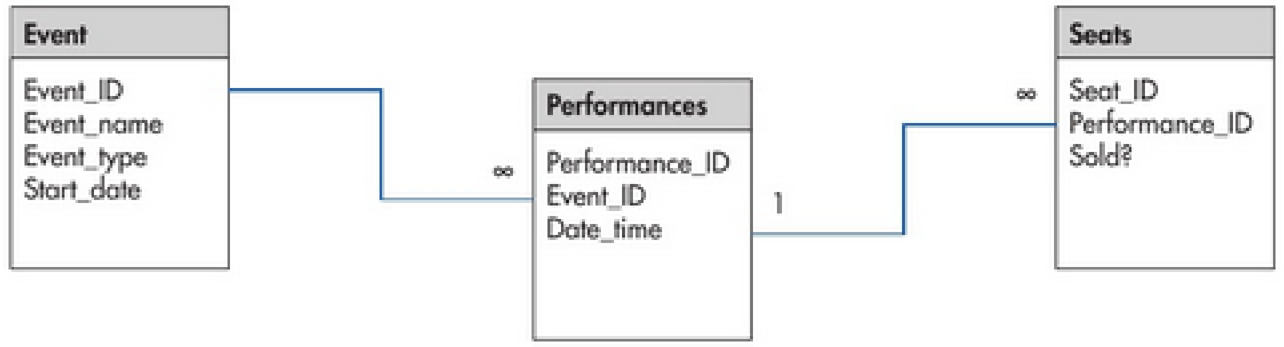
A theatre will put on various events, such as plays, musical performances, comedy shows, and so on. Each event will run for one or more nights (sometimes called a performance) and seats can be booked for each performance. In database terminology, the Events, Performance and Seats are called entities. Entities are likely to become the tables in the database, although you might need to add additional tables, for example, to resolve many-to-many relationships.

From this description you can see that one Event has many Performances, but a Performance can only be for one Event, so there is a standard one-to-many relationship. Figure 18.17 shows an ERD for this relationship with some typical fields added to each table.


Event
Event_ID
Event_name
Event_type
Start_date
1
∞
Performances
Performance_ID
Event_ID
Date_time

▶ Figure 18.17 An ERD showing the relationship between Event and Performance

Each performance in the theatre will have many seats that can be sold. The exact number of seats will be fixed by the capacity of the theatre, but the database would need to list and track the sales of all the seats for each performance. So, one performance has many seats and in terms of the availability and booking the seat, each seat must belong to a single performance. Therefore, this is once again a one-to-many relationship. Figure 18.18 shows the full ERD.


Event
Event_ID
Event_name
Event_type
Start_date
∞
Performances
Performance_ID
Event_ID
Date_time
1
∞
Seats
Seat_ID
Performance_ID
Sold?

▶ Figure 18.18 The three table ERD for the theatre booking application

This database application is a little different from the orders application you have looked at. With the orders application, records in the Orders and Order items tables are only created when an order is placed. With the theatre application, there is a fixed number of seats available for any performance, and the application needs to be able to record which seats have been sold and which are still available. Therefore, when a performance record is created, a corresponding set of seat records also need to be created (one record for each seat that is available in the theatre). Ideally an automated routine would be developed to add the seat records when a performance record is created.

SKILLS

ANALYSIS, PROBLEM SOLVING

ACTIVITY 3

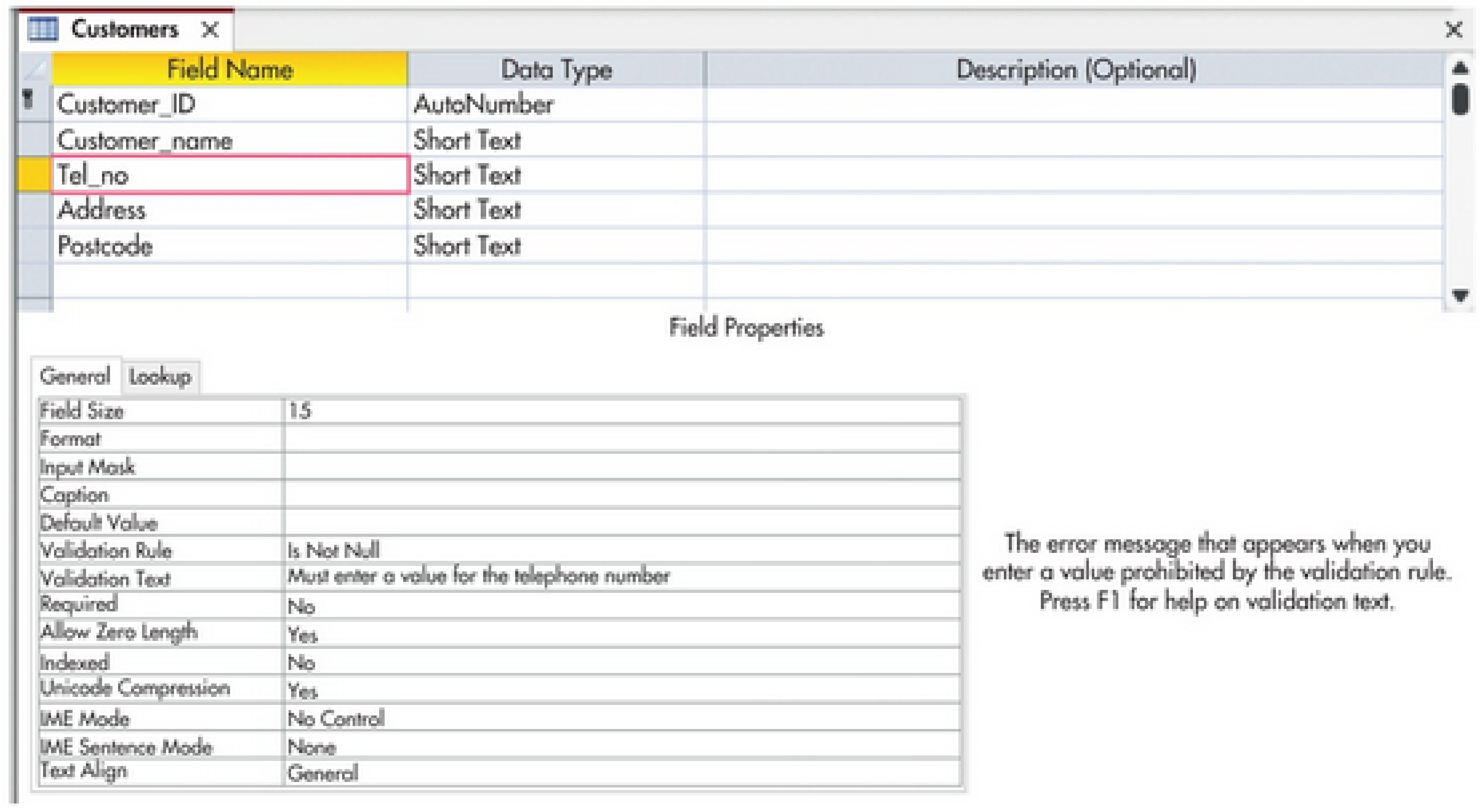
1. What is the difference between a one-to-many and a one-to-one relationship?
2. Many-to-many relationships cannot be represented in a relation database. How are they resolved?
3. A company that organises music festivals is designing a database to record details of people who attend the festivals and the festivals themselves. Each festival has many people attending, but some people will attend more than one festival. Is this a many-to-many relationship? How can this be resolved?
4. Each festival in the database described above has several different bands playing. How can this be represented in the database?
5. Create an outline design for the festivals database, adding some of the fields that might be needed and decide what datatype they should use. Draw an ERD for the database design.
6. In the theatre database described above what would the keys and foreign keys be for the tables? What fields might be needed for the tables? What datatypes should be used?

DATA ENTRY AND VALIDATION

It is important that the data added to a database is correct. If it is not, the database may produce incorrect results and fail to support the application properly. Therefore, the fields in a table should include **validation** wherever possible to prevent users from entering invalid data.

Key values. As you have seen, database records should have a key, such as a customer ID or order number. The value for these keys should not be entered by the user as they need to be unique. Where an Access AutoNumber datatype has been used, Access will create the unique sequential number for you. Where other codes are used, the data input form should calculate the correct code without the user having to do anything.

Presence check. This is a check that something has been entered in a field. The simplest way to do this is to use the validation rule in the field properties. Open the Design view of the table that has the field that you want to add a presence check to. Figure 18.19 shows the Customer table in Design view with a validation rule of 'Is Not Null' added to the Tel_no field.



Field Name	Data Type	Description (Optional)
Customer_ID	AutoNumber	
Customer_name	Short Text	
Tel_no	Short Text	
Address	Short Text	
Postcode	Short Text	

Field Properties

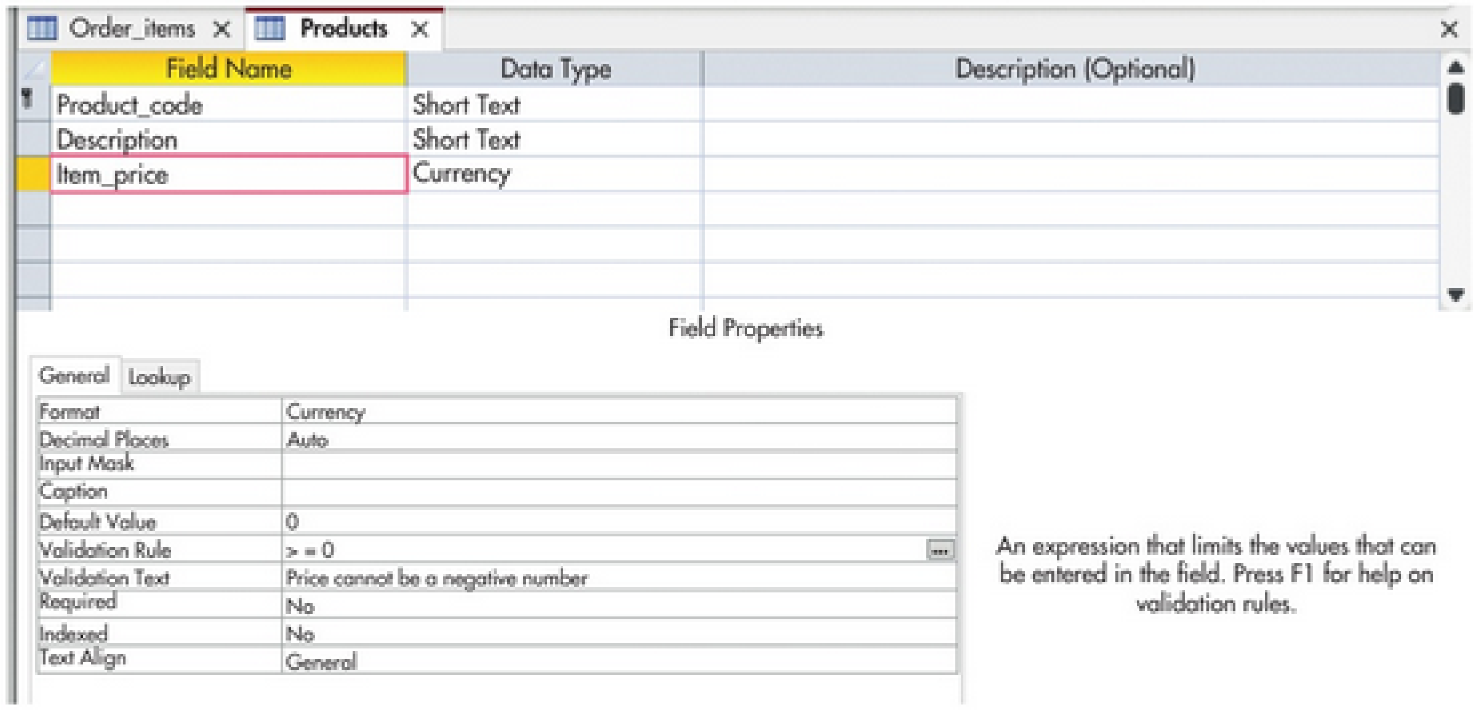
Field Size	15
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	Is Not Null
Validation Text	Must enter a value for the telephone number
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Text Align	General

The error message that appears when you enter a value prohibited by the validation rule. Press F1 for help on validation text.

▶ Figure 18.19 Adding a presence check validation rule

Along with the validation rule you should add validation text, which is the error message that appears if the validation rule is broken.

Range check. For some numeric fields there may be a limited range of values allowed. For example, if you have a field for a person's age then the range of values is really limited to 0 to 120. Remember that numeric fields can hold negative values so preventing their input by mistake is worth doing. A range check is added to the validation rule setting the same way as a presence check. Figure 18.20 shows a range check for the price field of the Product table to stop a negative value being entered.



Field Name	Data Type	Description (Optional)
Product_code	Short Text	
Description	Short Text	
Item_price	Currency	

Field Properties

Format	Currency
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	> = 0
Validation Text	Price cannot be a negative number
Required	No
Indexed	No
Text Align	General

An expression that limits the values that can be entered in the field. Press F1 for help on validation rules.

▶ Figure 18.20 Adding a validation rule to check a value is not a negative number

Strictly speaking, this is not a range check as there is no upper value check. An age field value check could use a validation rule such as >=0 AND <120. Validation rules can use the logical functions of AND and OR to combine rules.

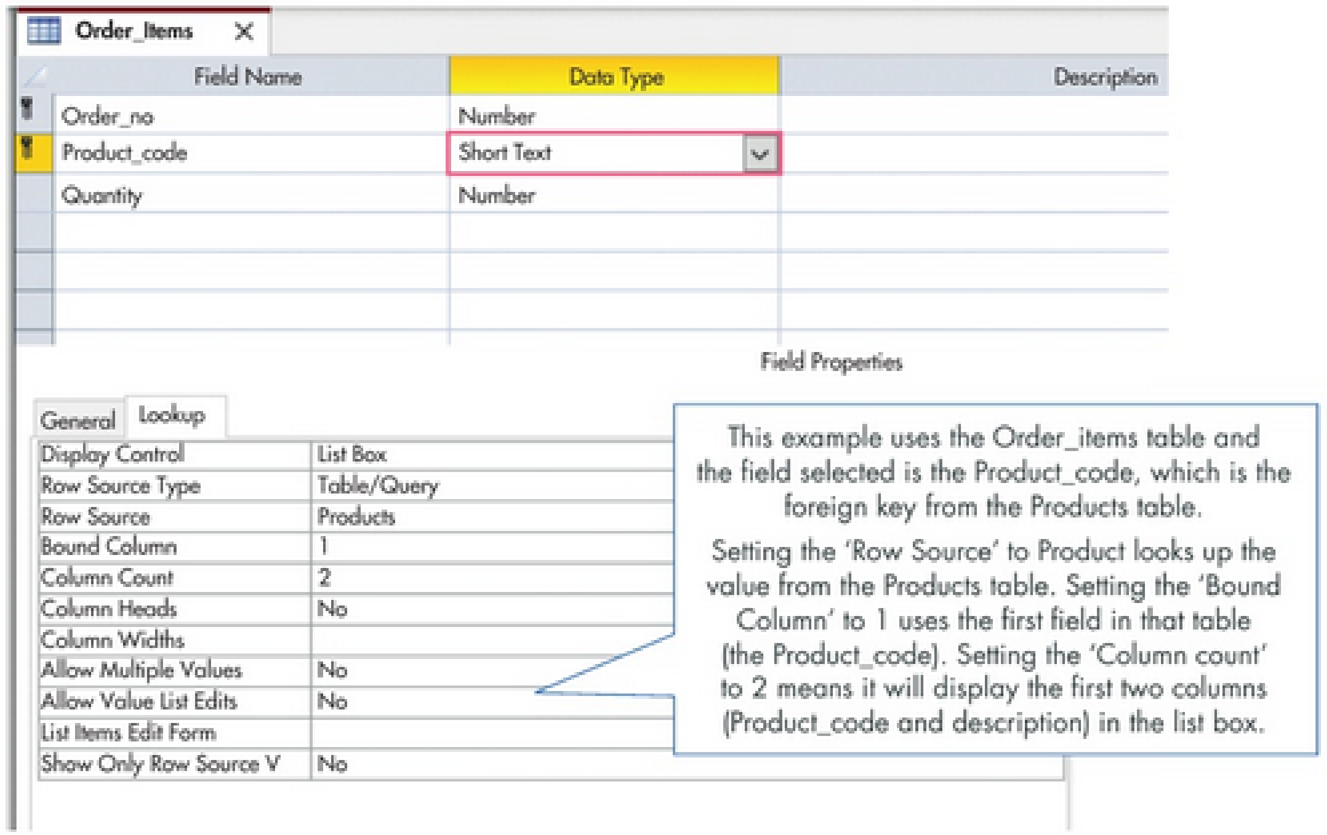
KEY TERM

lookup: a method of providing input data by either looking it up from another table or by providing a list

Lookup check. This is used to look up a value in another table.

When creating a record in the 'many' end of a one-to-many relationship, a foreign key value will need to be added to a new record. For example, with the Orders database you have been looking at, when an order is created a customer for the order must be selected from the Customers table so their customer number can be inserted as a foreign key in the Order record. There are several ways this can be done, but one way is to provide a drop-down box with a list of valid customer numbers (and names) taken from the Customer table. The user can then only select a valid customer number from the list. Remember that with Referential Integrity turned on, the foreign key used in the Orders table must be a valid value from the Customer table, otherwise the record will not save.

To add a lookup to a field, you need to open the table in Design view then click in the field you want to add the lookup to, and choose the Lookup tab in the field properties section of the Design view (Figure 18.21). In the 'Display Control' property you need to use the drop-down box to select List Box rather than the default setting of Text Box.



Field Name	Data Type	Description
Order_no	Number	
Product_code	Short Text	
Quantity	Number	

Field Properties

Display Control	List Box
Row Source Type	Table/Query
Row Source	Products
Bound Column	1
Column Count	2
Column Heads	No
Column Widths	
Allow Multiple Values	No
Allow Value List Edits	No
List Items Edit Form	
Show Only Row Source V	No

This example uses the Order_items table and the field selected is the Product_code, which is the foreign key from the Products table.

Setting the 'Row Source' to Product looks up the value from the Products table. Setting the 'Bound Column' to 1 uses the first field in that table (the Product_code). Setting the 'Column count' to 2 means it will display the first two columns (Product_code and description) in the list box.

▶ Figure 18.21 Using a lookup to select a value from another table

List check. A list check is similar to a lookup check in that it provides a list box in the field for users to select items from, but rather than look them up in another table, the values are listed in the table design, so need to be fixed values that don't change. This is done using the list box option on the field properties shown above, but rather than choosing a Row source type of `Table/Query`, select the `Value List` option. The Row source property is then set to the list of values you want to appear in the list box.

Format (picture) check. With text fields you can use an input mask that forces the input to follow a certain pattern. The input mask property is listed in the field properties in Table Design view. Input masks will not let the user type an invalid character, and they can also do things like making all text capital letters. For example, in the Orders database you have been looking at, the products use a product code of one capital letter and three digits, for example A123. The input mask to force data input to that field to follow the format is:

>L000

Where the '>' symbol makes text uppercase, 'L' is a letter and '0' is a digit. The other codes used to define input masks are:

0 – digit is required
9 – optional digit
– optional digit, plus or minus sign or space
L – letter is required
? – optional letter
A – letter or digit is required
a – optional letter or digit
& – characters or space required
C – optional characters or spaces
> – all following characters will be in uppercase
< – all following characters will be in lowercase

An example could be an input mask for a telephone number. The following mask would be suitable for a 10 digit telephone number:

(000) 000-0000

The first three digits would be enclosed in brackets (there is no need to type the brackets, they are automatically inserted). After the second set of three digits, a dash is automatically inserted.

Another example could be to ensure people's names start with a capital letter. The input mask:

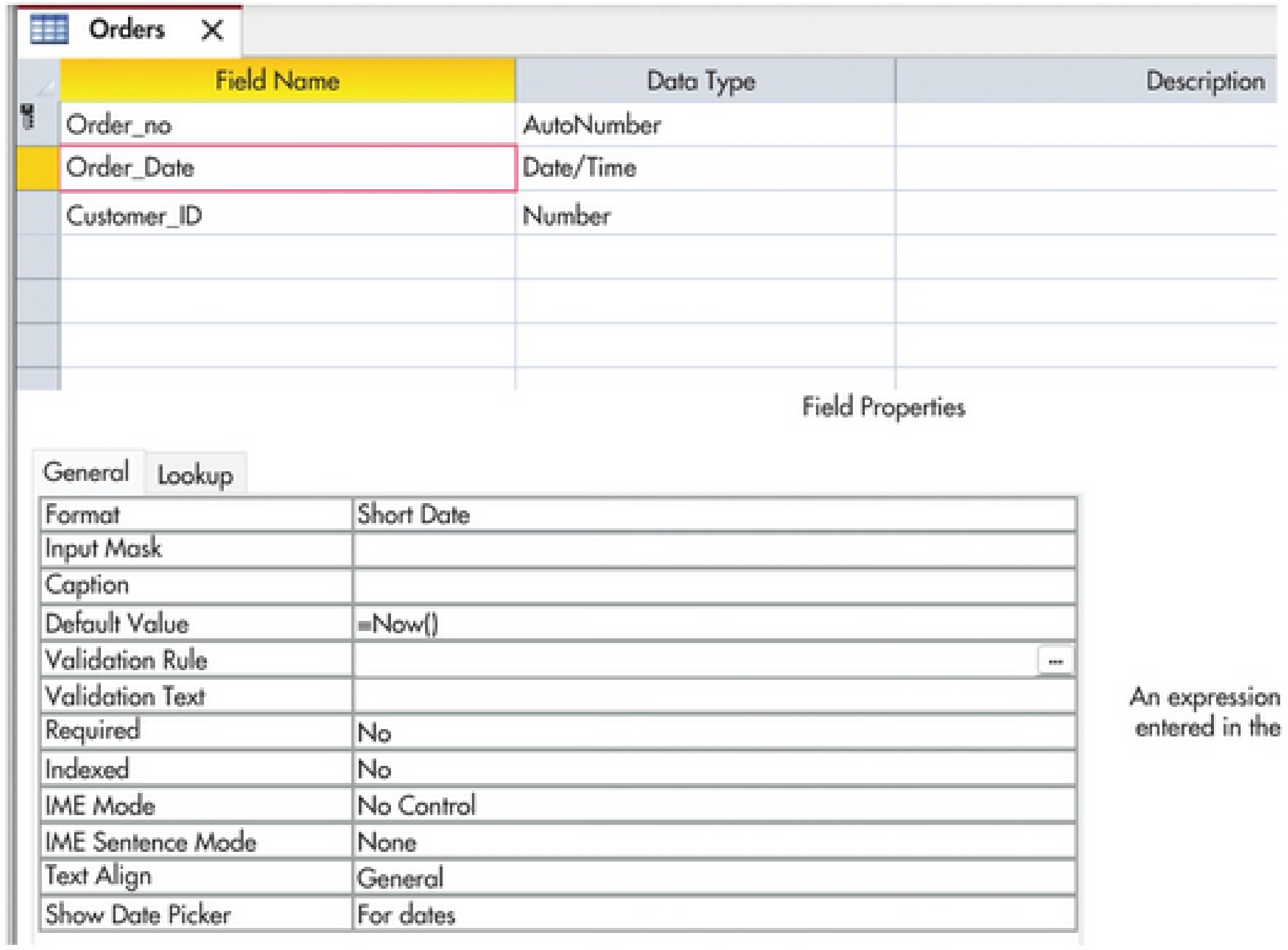
>L<L???????????????

would automatically capitalise the first letter of a name.

Note that input masks can be added to form fields, but if you apply an input mask to the field properties in Design view, that won't apply to the form field.

Length of data check. As mentioned earlier you should set suitable lengths for fields, but if a field has a fixed length (so for example it must always be 4 characters and 4 characters only), then an input mask can be used.

Default values. In some cases, you may want to add a default value to a field. This can make data entry easier in situations where there is a 'most often used' value for a field. For example, in the Orders database you have been looking at, the Orders table has a field for the order date. It is likely that the order will be recorded in the database on the day it is ordered. So rather than requiring the user to enter the current date, the default value for that field could be set to today's date. This can be done using the `now()` function in the default value property for the field. This is shown in Figure 18.22.



Field Name	Data Type	Description
Order_no	AutoNumber	
Order_Date	Date/Time	
Customer_ID	Number	

Format	Short Date
Input Mask	
Caption	
Default Value	=Now()
Validation Rule	
Validation Text	
Required	No
Indexed	No
IME Mode	No Control
IME Sentence Mode	None
Text Align	General
Show Date Picker	For dates

▶ Figure 18.22 Setting a default value to the current date

Error messages. You should always add a clear and meaningful message in the Validation Text property when you have a Validation rule. Otherwise, the default message will appear if the rule is broken, which is not user friendly. The message should make it clear what format the input should be in so the user can resolve the problem. Care should be taken with input masks as these will not allow the user to type characters that the mask does not allow, and they do not display any error message. You need to check carefully that the input mask allows all possible valid entries.

SKILLS
ANALYSIS

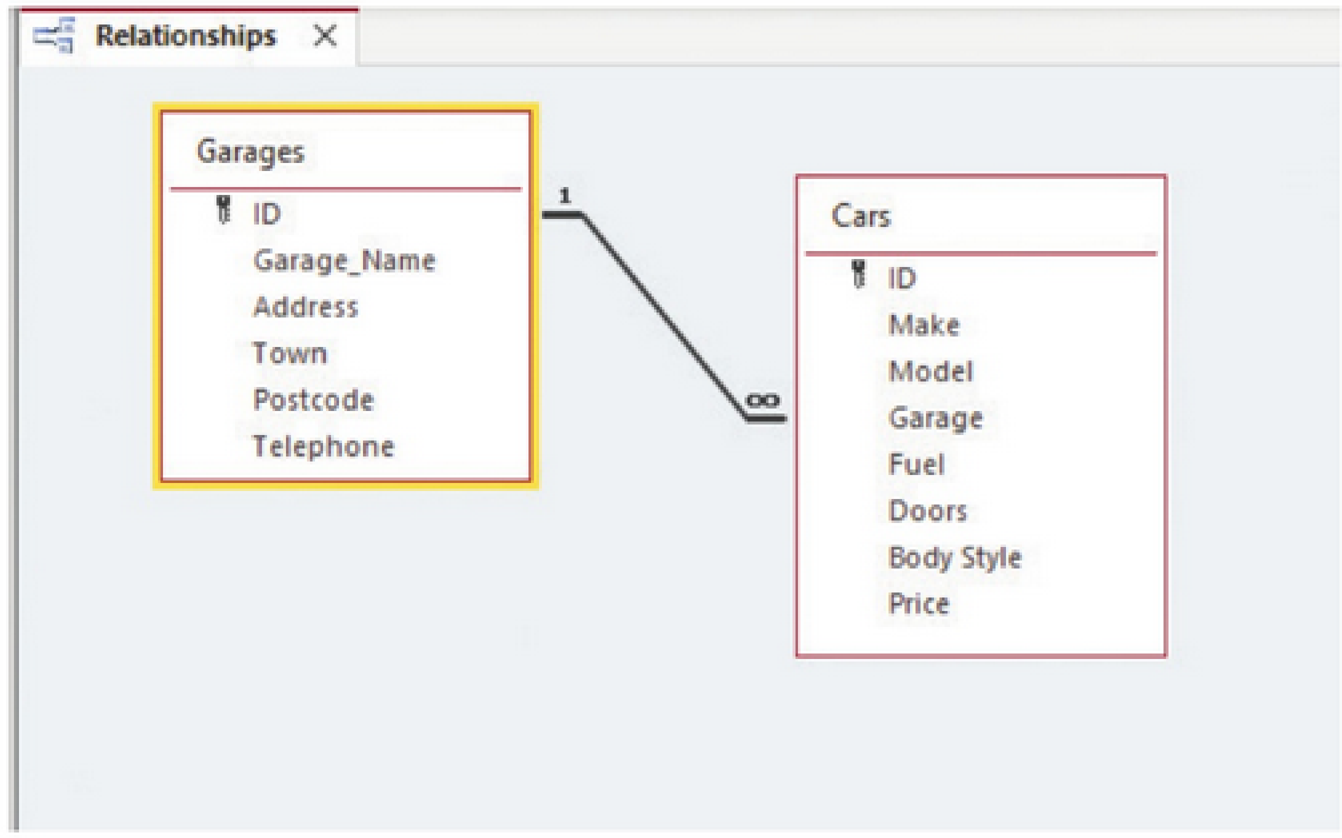
ACTIVITY 4

1. An address has a postal code field. The code is always in the format of two capital letters followed by two numbers, a space then one number and two capital letters, for example, PH21 9EX. What input mask would make an input follow this pattern?
2. Using one of the databases you have already created tables for (orders, festivals or theatre databases), add validation such as presence checks or lookup checks to the fields where relevant.

IMPROVING THE QUALITY OF DATA ENTRY

You can, of course, type new records (and amend existing ones) using the Datasheet view in Access, but to develop a true database application you need to create user-friendly forms to allow the users to create new records. Forms can include features (which Access calls 'controls') such as help, drop-down lists, radio buttons, check boxes and other automated features. As you can add validation to both the table properties and the form, you should decide how the table will be used and data entered. It is not a good idea to add validation to both the table properties and the form as this can cause conflicts and confusion. If you are creating an end-user database application that will be accessed by the forms, then put the validation on the forms. See below for how to create a sophisticated form with Access.

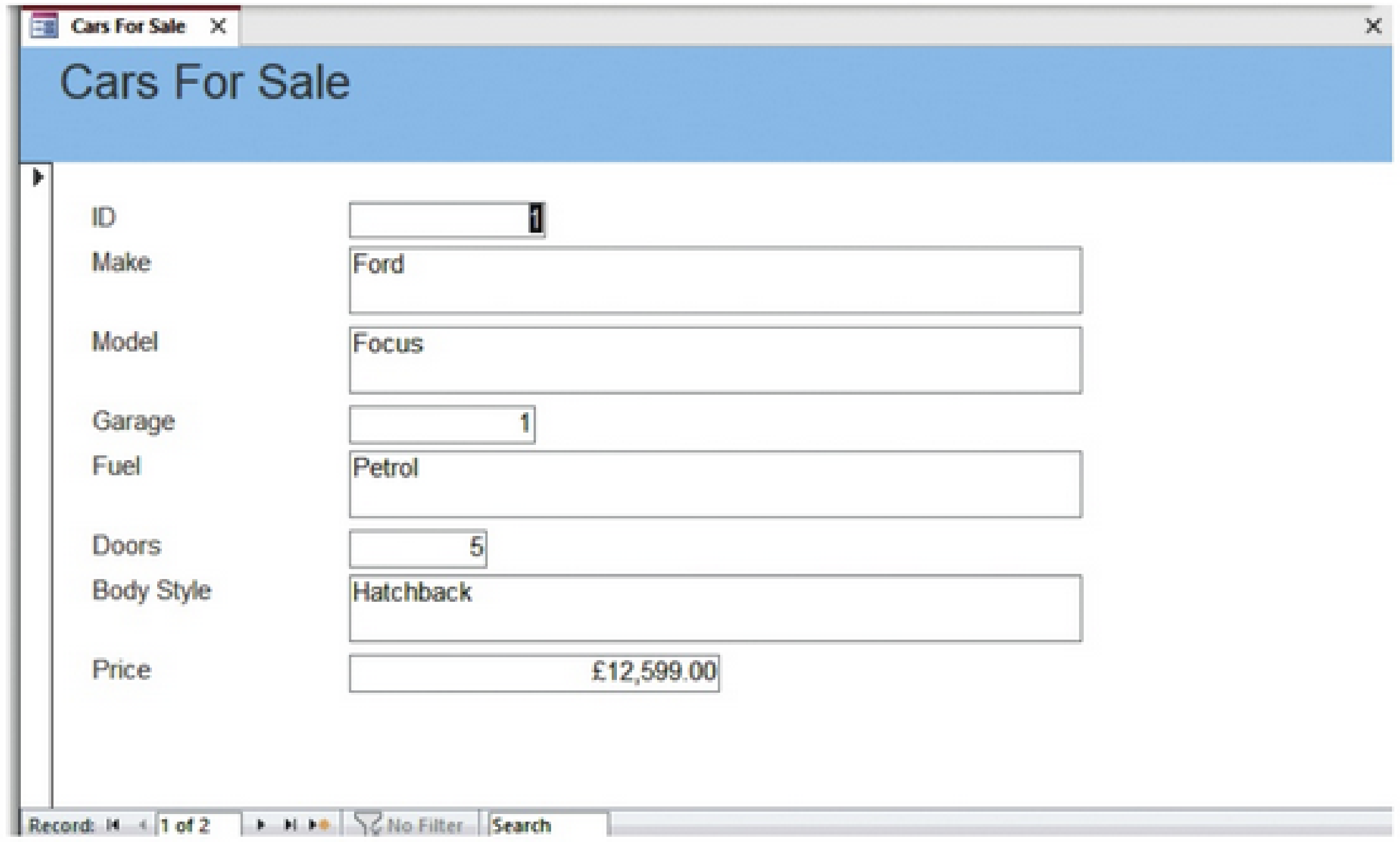
To show all the different features, look at this database, which records details of cars that a dealer has for sale in its garages. The ERD for this database is shown in Figure 18.23.



▶ Figure 18.23 The ERD for the car sale database

The relationship between the tables is that one garage can have many cars for sale.

The easier way to create a form in Access is to use the Form Wizard, which guides you through creating a form; you can then customise the form that the Wizard creates. To create a basic form for the cars table, have that table selected, then from the Create menu, click the Form Wizard button. Add all the fields for the table to the form then choose a 'Columnar' Layout, change the title of the form to 'Cars For Sale', then click finish to create the form. This creates a form like the one shown in Figure 18.24. Using the record selectors at the bottom of the form, you can scroll through the records and add a new record.



Cars For Sale

ID	1
Make	Ford
Model	Focus
Garage	1
Fuel	Petrol
Doors	5
Body Style	Hatchback
Price	£12,599.00

▶ Figure 18.24 A form created with the Form Wizard

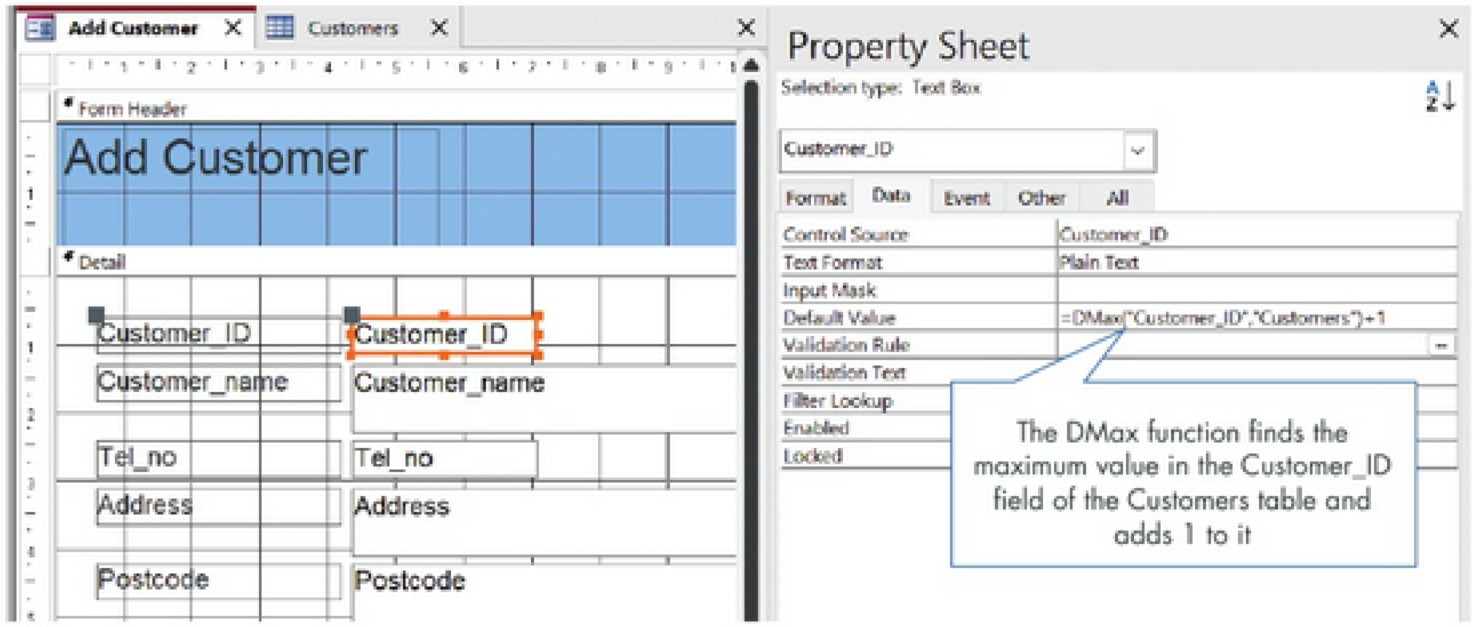
This kind of 'multi-purpose' form is not best suited for an end-user database application. It is better to have a form for data entry (adding new records) and another that allows the user to search for a record. Therefore, you should change this form to a data entry form and add features to make it easier to use.

To make this a data entry form simply go to the Form Design view, make sure the Property sheet is showing (click the Property sheet on the right of the toolbar if it isn't visible). Then choose the Data tab on the property sheet and change the Data Entry property to Yes.

The primary key field

The other change you should make is to disable the ID field on the form. The ID field is the primary key of the table and its datatype is set to AutoNumber. The user should not therefore need to enter anything in this field, Access will create sequential unique numbers for the key. Click on the files and then in the Property Sheet Data tab, change the Enable property to No and the Locked property to Yes.

If an AutoNumber datatype has not been used for a primary key field, you will need a way to make sure that the primary key is unique. If the primary key is numeric, then one way to do this is to take the highest key value and simply add one to it. Access provides a function called DMax, which will find the highest value in a numeric field. You can therefore set the default value for the key field by adding one to the DMax value. DMax takes two parameters, the first one is the field to find the maximum value in (which would be the key field itself) and the second is the table the field is in, which is the same table as the key belongs to. Figure 18.25 shows an example.



Selection type: Text Box

Customer_ID

Control Source	Customer_ID
Text Format	Plain Text
Input Mask	
Default Value	=DMax("Customer_ID","Customers")+1
Validation Rule	
Validation Text	
Filter Lookup	
Enabled	
Locked	

The DMax function finds the maximum value in the Customer_ID field of the Customers table and adds 1 to it

▶ Figure 18.25 Using the DMax function to find the highest key value

Selecting the foreign key value

The next task is to provide a drop-down box to select the garage where the car is located. This is the foreign key from the garages table, so it must be a valid Garage ID. Access provides two ways of presenting a list that the user can select from:

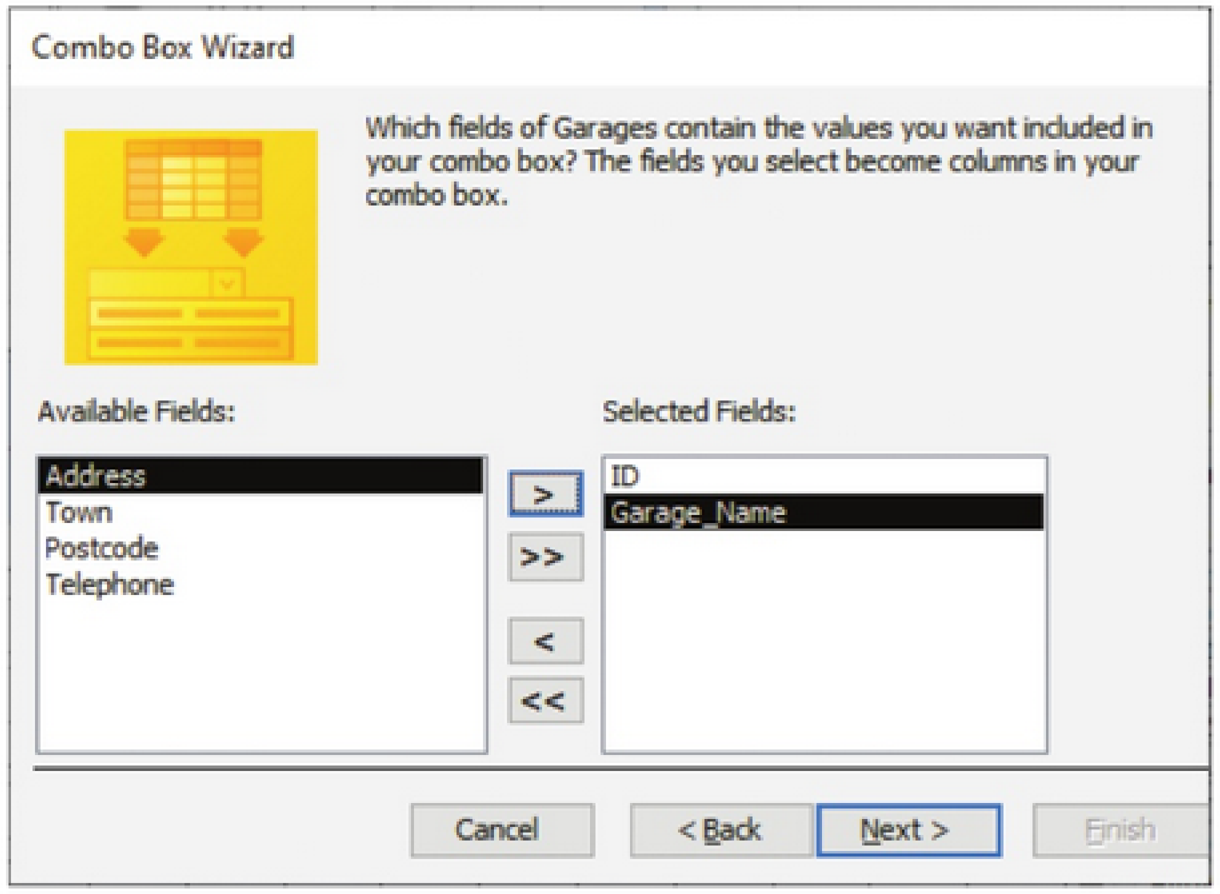
Combo box – this provides a list of options that is hidden until the user clicks on the control, and then the list drops down. You will use a combo box in this example.

List box – this provides a list that is always displayed. A scroll bar is added if the full list will not fit in the box that you create.

To create these features, in the Form Design view delete the existing field for the garage. Then, from the Form Controls section of the Form Design tool bar choose the Combo Box control and drag out a combo box in the space where the garage field was. This will open the Combo Box Wizard. Work through the pages of the Wizard as follows:

1. Make sure that on the first Wizard page you have the first option selected to get the values from another table or query. Then click Next.
2. On the second page, the 'Table: Garages' option should be selected. Click Next again.

Figure 18.26 shows how the form will now look.


Combo Box Wizard

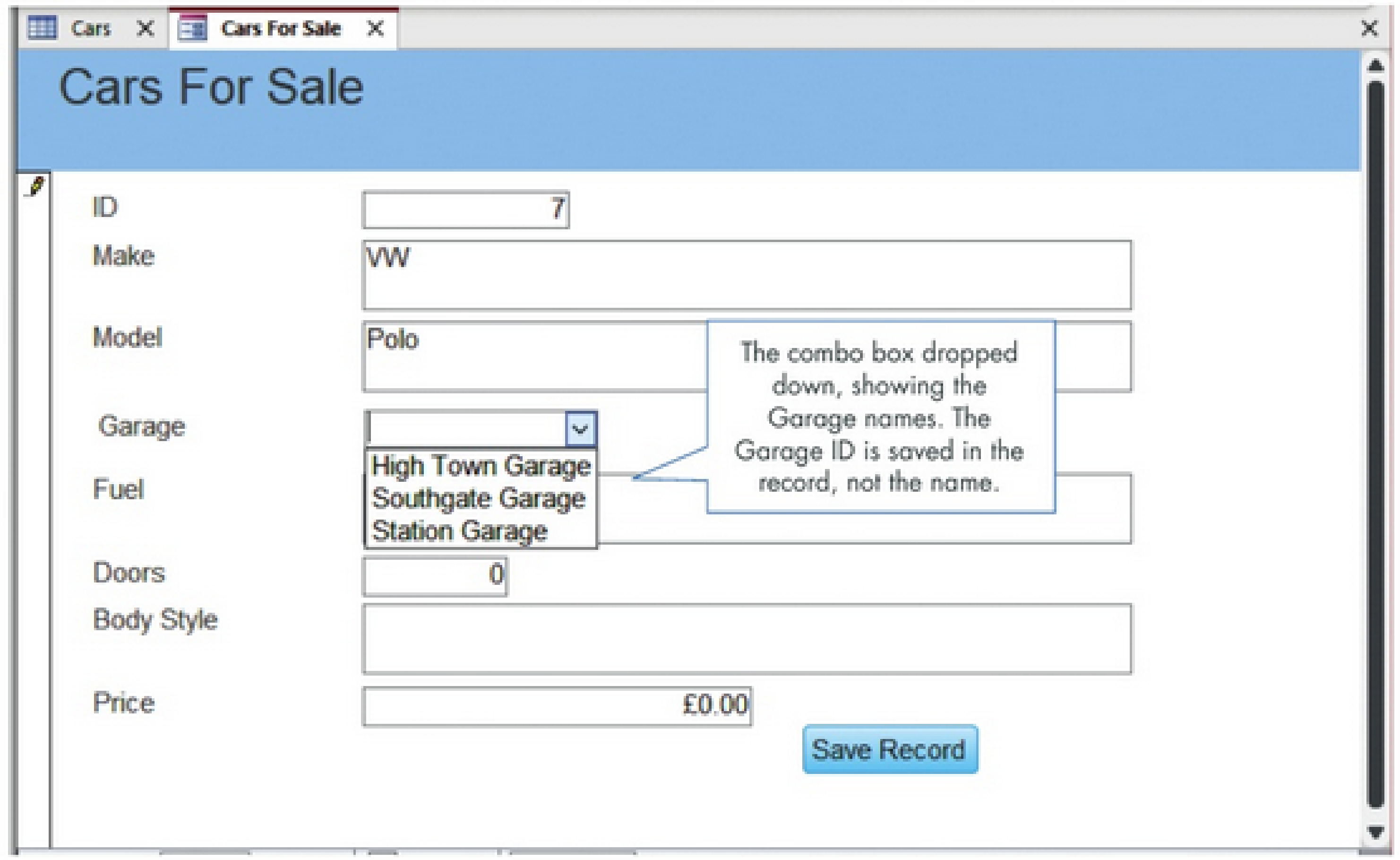
Which fields of Garages contain the values you want included in your combo box? The fields you select become columns in your combo box.

Available Fields:
Address
Town
Postcode
Telephone
>
>>
<
<<
Selected Fields:
ID
Garage_Name
Cancel
< Back
Next >
Finish

▶ Figure 18.26 Selecting the fields to display in the combo box

3. On this page, the ID and the Garage Name have been added to the fields to be displayed. The ID is the value you want to use for this field, but having the Garage Name will help the user to see if they have the correct garage. Click the Next button.
4. The Wizard will ask if you want the values sorted, which in this case is not necessary, so just click Next.
5. The next page allows you to set the width of the columns in the combo box. Leave the first field hidden as recommended (this is the key field). Click Next.
6. This page asks you where you want to save the selected value from the combo box. Select Store that value in this field and choose the Garage field from the drop down. Click Next.
7. Finally, give the combo box a label of Garage and click Finish.

The form will now have a combo box from which you can select the garage where the car is located, and it will place the ID of that garage in the foreign key field when the car record is saved. Figure 18.27 shows the form with the combo box.


Cars
Cars For Sale
Cars For Sale
Fuel
Body Style
Save Record

ID 7
Make VW
Model Polo
Garage
High Town Garage
Southgate Garage
Station Garage
Doors 0
Price £0.00

The combo box dropped down, showing the Garage names. The Garage ID is saved in the record, not the name.

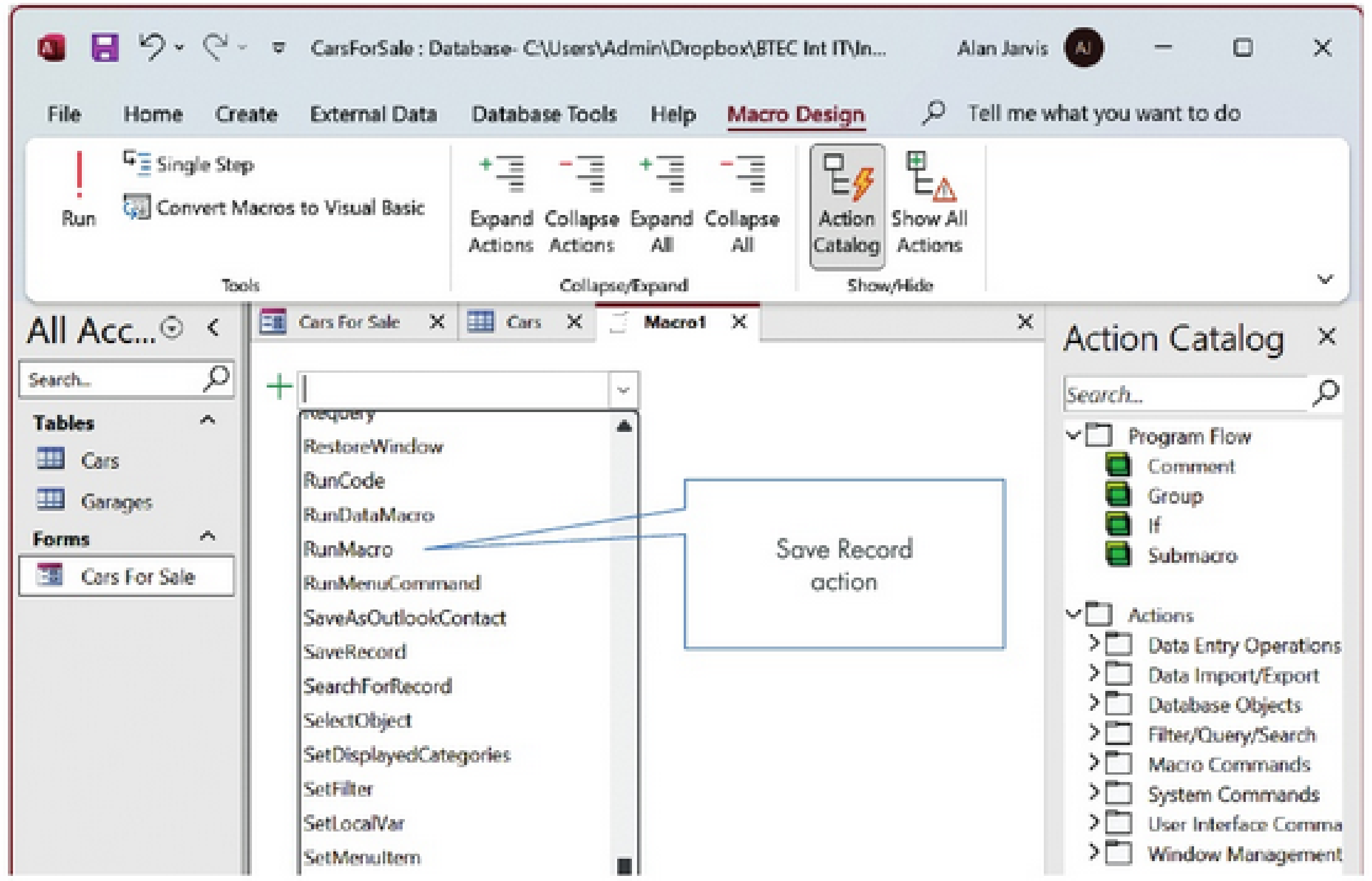
▶ Figure 18.27 The Cars For Sale form with the complete combo box

KEY TERM

macro: a series of actions that allow you to automate database procedures such as saving a record

Adding a save button with an automated process

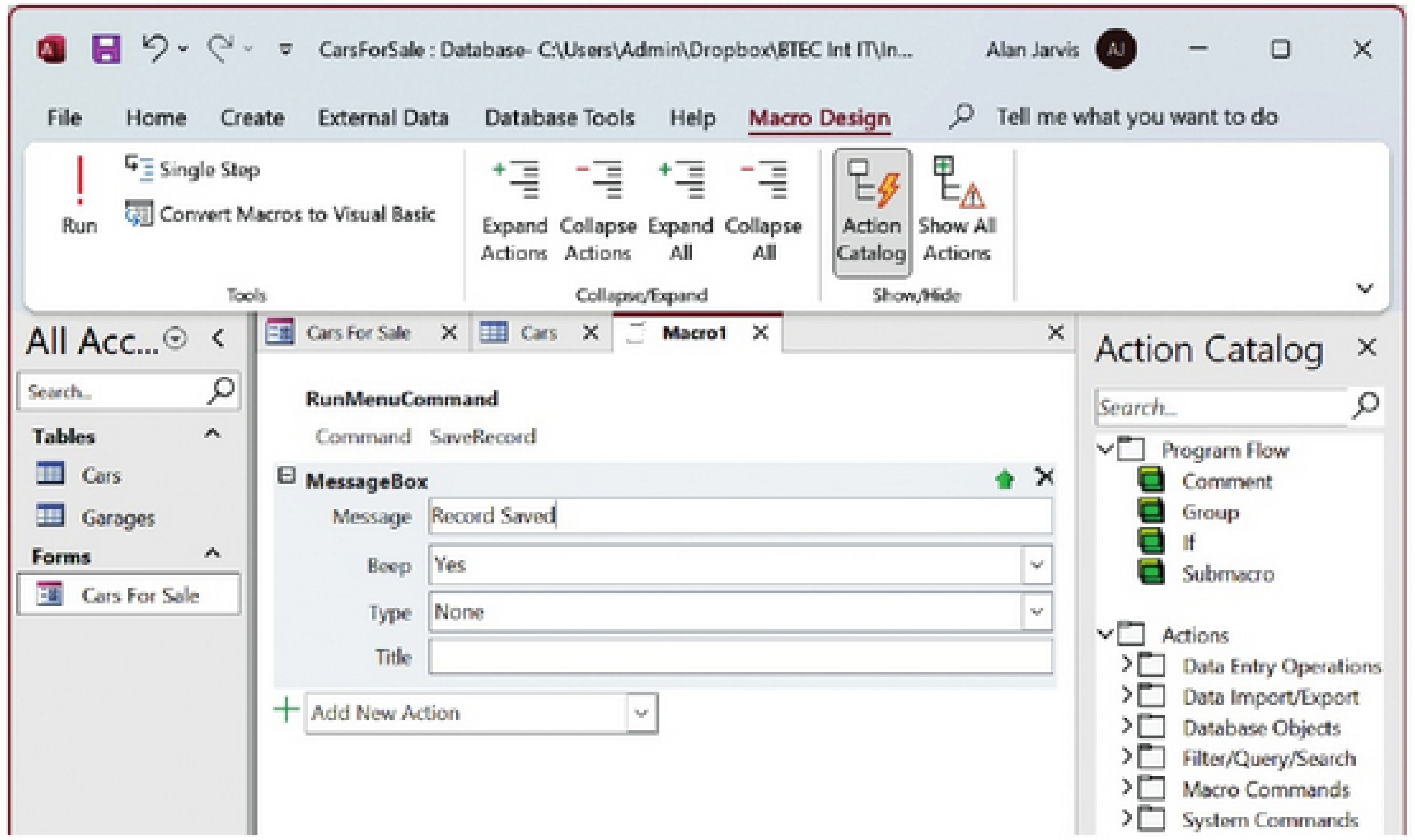
Another improvement that can be made to this form is to add a save button to make it easy for the user to save the record they have entered. There are several ways to do this. You could use the Command button wizard, but creating an automated process with a macro gives greater flexibility and allows you to add additional features. To create a macro, go to the Create menu, then click Macro on the right side of the toolbar and the Macro Window will open. Click the drop down box, scroll down through the list of actions and choose Save Record, as shown in Figure 18.28.


CarsForSale : Database- C:\Users\Admin\Dropbox\BTEC Int IT\In...
Alan Jarvis
AJ
File
Home
Create
External Data
Database Tools
Help
Macro Design
Tell me what you want to do
Run
Single Step
Convert Macros to Visual Basic
Tools
Expand Actions
Collapse Actions
Expand All
Collapse All
Collapse/Expand
Action Catalog
Show All Actions
Show/Hide
All Acc...
Search...
Tables
Cars
Garages
Forms
Cars For Sale
Cars For Sale
Cars
Macro1
Requery
RestoreWindow
RunCode
RunDataMacro
RunMacro
RunMenuCommand
SaveAsOutlookContact
SaveRecord
SearchForRecord
SelectObject
SetDisplayedCategories
SetFilter
SetLocalVar
SetMenuItem
Action Catalog
Search...
Program Flow
Comment
Group
If
Submacro
Actions
Data Entry Operations
Data Import/Export
Database Objects
Filter/Query/Search
Macro Commands
System Commands
User Interface Comma
Window Management

Save Record action

▶ Figure 18.28 Adding the Save Record action to macro

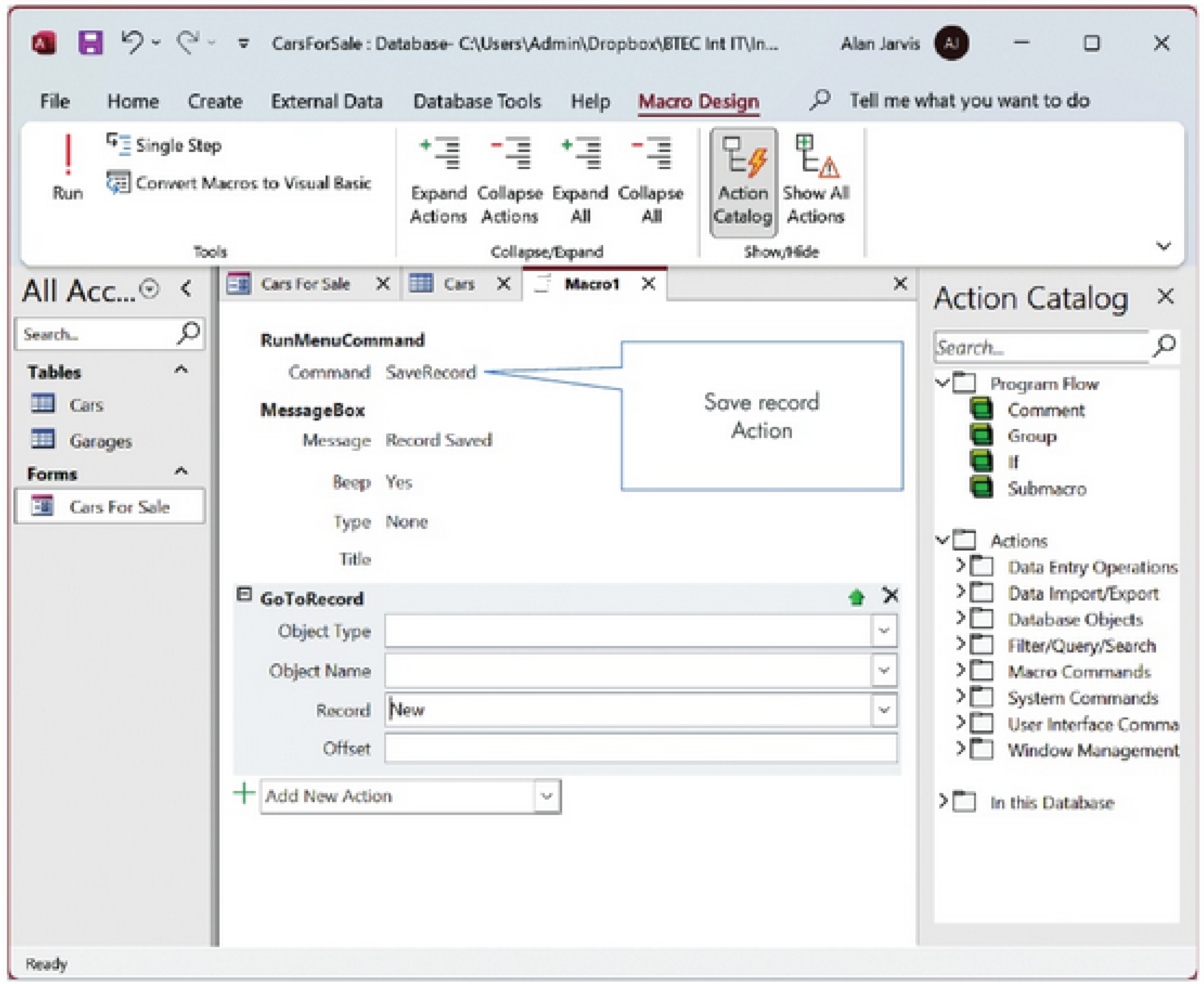
Then, since it would be nice to provide confirmation to the user that the record has been saved, add the action Message Box and include a message such as Record Saved. See Figure 18.29.


CarsForSale : Database- C:\Users\Admin\Dropbox\BTEC Int IT\In...
Alan Jarvis
File
Home
Create
External Data
Database Tools
Help
Macro Design
Tell me what you want to do
Run
Single Step
Convert Macros to Visual Basic
Tools
Expand Actions
Collapse Actions
Expand All
Collapse All
Collapse/Expand
Action Catalog
Show All Actions
Show/Hide
All Acc...
Search...
Tables
Cars
Garages
Forms
Cars For Sale
Cars For Sale
Cars
Macro1
Add New Action
Action Catalog
Search...
Program Flow
Comment
Group
If
Submacro
Actions
Data Entry Operations
Data Import/Export
Database Objects
Filter/Query/Search
Macro Commands
System Commands

RunMenuCommand
Command SaveRecord
MessageBox
Message Record Saved
Beep Yes
Type None
Title

▶ Figure 18.29 Add a message box to the macro

Finally, the macro should clear the form so that another record can be added. This can be done by adding the action GoToRecord and selecting New in the record field. See Figure 18.30.


CarsForSale : Database- C:\Users\Admin\Dropbox\BTEC Int IT\In...
Alan Jarvis
File
Home
Create
External Data
Database Tools
Help
Macro Design
Tell me what you want to do
Run
Single Step
Convert Macros to Visual Basic
Tools
Expand Actions
Collapse Actions
Expand All
Collapse All
Collapse/Expand
Action Catalog
Show All Actions
Show/Hide
All Acc...
Search...
Tables
Cars
Garages
Forms
Cars For Sale
Cars For Sale
Cars
Macro1
Add New Action
Action Catalog
Search...
Program Flow
Comment
Group
If
Submacro
Actions
Data Entry Operations
Data Import/Export
Database Objects
Filter/Query/Search
Macro Commands
System Commands
User Interface Comma
Window Management
In this Database
Ready

RunMenuCommand
Command SaveRecord
MessageBox
Message Record Saved
Beep Yes
Type None
Title
GoToRecord
Object Type
Object Name
Record New
Offset

Save record Action

▶ Figure 18.30 Adding an action to display a new record

The macro is now complete, so click the Save button in the top left of the window and give it a name, such as SaveCar.

Now open the form in Design view again, go to the Controls section of the Form Design toolbar and create a Command Button. In the Button Wizard choose the Miscellaneous category and the Run Macro action. Select the SaveCar macro you just created and add a suitable message to the button.

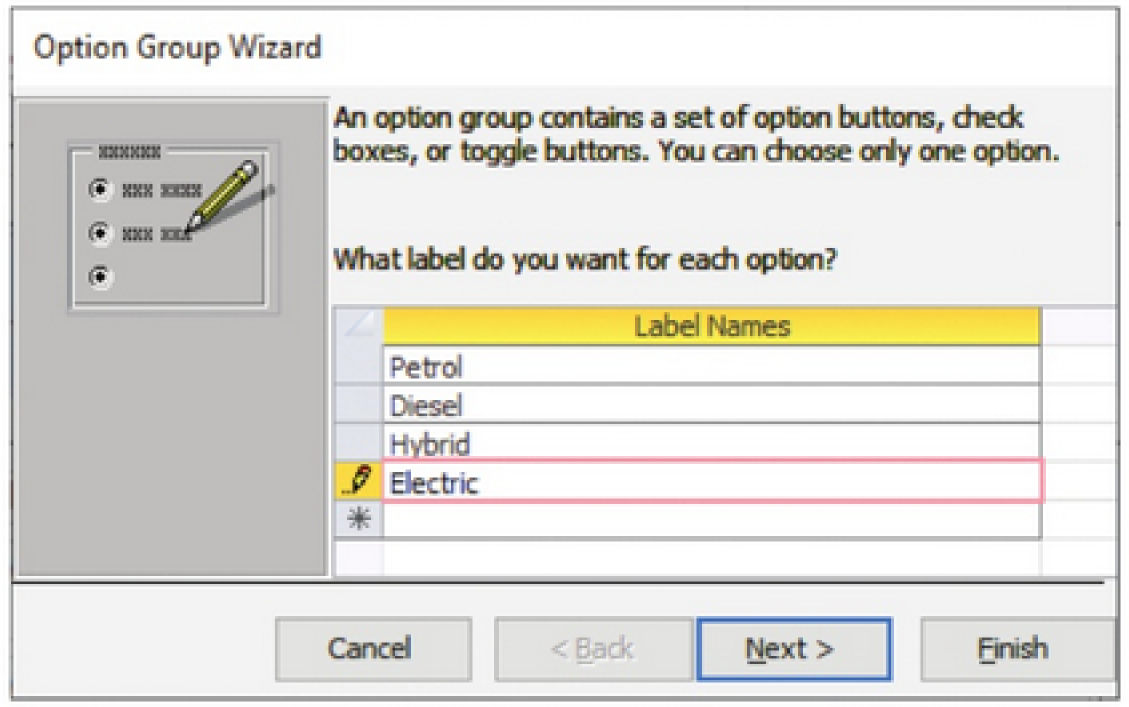
The button will now save the record that has been entered, display a confirmation record and display a new blank record.

As well as a Save button, a Cancel button should also be provided. Just closing the form will not work as the record will be saved when the form closes. To cancel an update, you need to create a macro that uses the RunMenuCommand action and select the UndoRecord command. Save this macro and attach it to a Command button labelled Cancel as described above.

Adding option buttons

The form has a number of fields on it for free text entry, but there are only a limited number of options, so these would be easier to use if they were replaced by list boxes, option buttons or check boxes. The fuel that a car can use for example is limited to petrol, diesel, hybrid or electric, so an option (radio) button set would be an ideal way for the user to select from these options.

To create an option button set, first delete the field for the fuel in the Form Design view. Then from the Form Control section, select then drag out an Option Button Set. This will open the Option Group Wizard. In the first page enter the different options that need to have buttons. Figure 18.31 shows an example for the fuel type.


Option Group Wizard
Cancel
< Back
Next >
Finish

An option group contains a set of option buttons, check boxes, or toggle buttons. You can choose only one option.

What label do you want for each option?

Label Names
Petrol
Diesel
Hybrid
Electric

▶ Figure 18.31 Entering the label names for option buttons

Add the different fuel types and click Next.

Leave the default choice set and click Next.

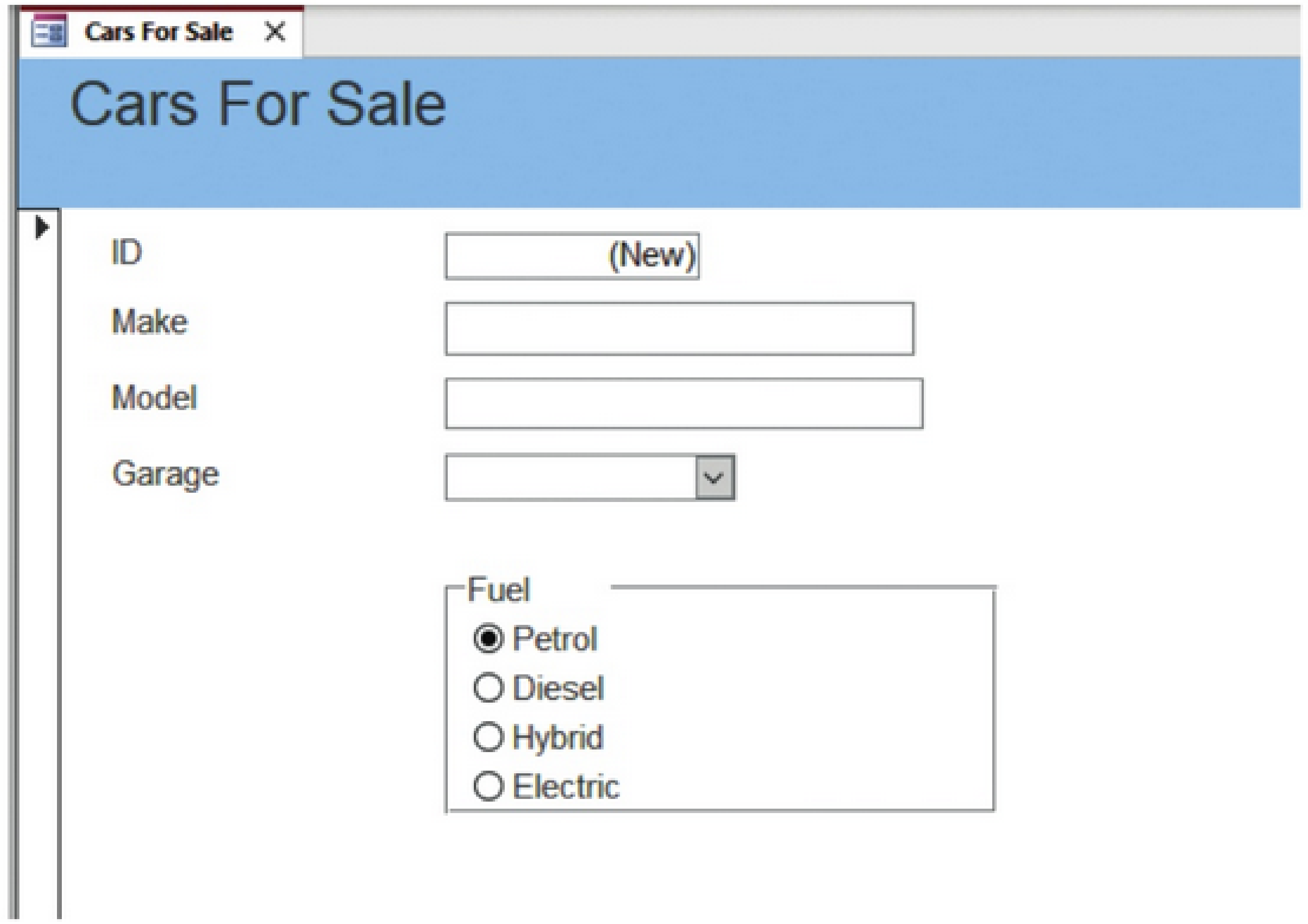
This page asked what value you want to assign to each option. You can only assign numeric values so leave these as they are. Click Next.

On this page set the field where the value will be stored to the Fuel field, click Next.

Select the type of button you want. These should be Option buttons.

Finish the Wizard and the Option Group should be created. You might need to move the other field around to make room for it.

The Option Group can now be used to select the fuel type. Figure 18.32 shows the completed form.



ID: (New)

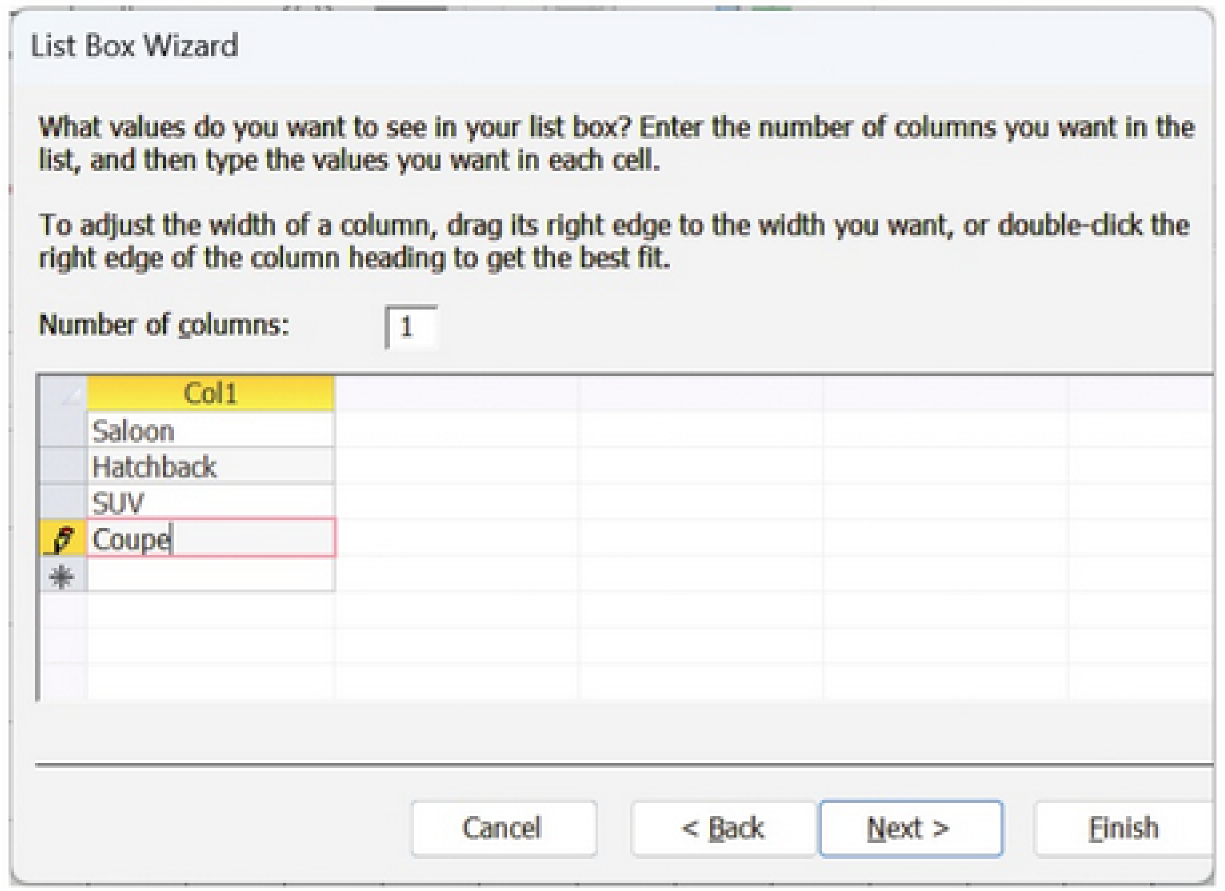
Fuel:
● Petrol
○ Diesel
○ Hybrid
○ Electric

▶ Figure 18.32 The form with an option group added

Using the option buttons to select the fuel type will mean that a number (1 to 4) will be saved in the Fuel field rather than the actual fuel type.

There are several other fields in this table that would benefit from list boxes or radio buttons. Cars can only have two, three, four or five doors so this could be selected using a list box. The car's body style should also be restricted to a given list of options (such as Hatchback, Coupe, SUV, etc.), which could be selected using radio buttons or a list box.

Adding a list box is very similar to adding a combo box. This time, rather than choosing a table to look up the values for the list, you need to choose to add them yourself. Figure 18.33 shows list options being added in the List Box Wizard.



What values do you want to see in your list box? Enter the number of columns you want in the list, and then type the values you want in each cell.

To adjust the width of a column, drag its right edge to the width you want, or double-click the right edge of the column heading to get the best fit.

Number of columns: 1

Col1
Saloon
Hatchback
SUV
Coupe

▶ Figure 18.33 Adding values to a list box

Unlike an option button set, using a list box will add the actual value to the field rather than a number.

Providing help

The end-user form should provide some guidance and help for this user. This can include:

- Brief written instructions on the form.
- A way of indicating which fields are mandatory. This is often done by placing an asterisk (*) in the labels for mandatory fields.
- Control tips. These are small text boxes that pop up when the mouse hovers over the label for a control. These can be used to provide additional instruction for the user to help them complete the field. The control tip text can be set in the properties for the field.

Evaluating validation techniques

The data accuracy techniques you apply to a table or form have two main goals. One is to ensure, as far as possible, that the data that goes in the database is correct and in the right format. Incorrect data can have all sorts of consequences, such as parcels sent to the wrong address, incorrect management decisions made, financial errors and so on. Some of the consequences could be serious for the business and for individuals. The other goal is to create an easy-to-use user interface that allows users to enter correct data efficiently. When evaluating the techniques that have been used, you should keep these two goals in mind.

The key question to ask is does the validation applied reject all possible invalid entries and accept all correct ones? This question can only be fully answered by going back to the original user requirements and having a clear understanding, for each field in question, of what valid and invalid data is. For some fields this is quite straightforward, but the definition of valid and invalid must be clear. For example, a field that takes someone's age might require that they are over 18. But is that 18 and over or over 18 (i.e. 18 is invalid but 19 is valid)?

Poorly implemented validation can cause user frustration. Users may find they cannot enter data that they think is valid, or they do not understand why the data they are entering is being rejected. Clear error messages can help avoid these problems. Validation also needs to be carefully tested to ensure a field really does accept valid data and reject only invalid data.

SKILLS
ANALYSIS, PROBLEM SOLVING

ACTIVITY 5

1. Earlier you looked at a database for a theatre to record details of events, performances and seats. Create a version of this database and add a form that can be used to create performances. Add a combo box to the form to allow the user to select the event to link the performance to.
2. The theatre database has an Events table. Create a form to add new events. The table should have a field called 'Event Type' or similar. The only values that should be allowed are 'Play', 'Music', 'Comedy' or 'Other'. Create a radio (option) button set on the form to allow the user to select the event type.

THINKING LIKE AN IT SPECIALIST

Database design is quite a complex subject. The examples given in this chapter have been fairly simple and it's been quite easy to work out how to split data into related tables by considering the application and the relationships between the data. However, many real-world applications are much more complex than this and it can be difficult to work out how to split data into related tables. As you saw in Unit 3, there is a formal mathematical method called 'normalisation' that has been developed to help do this. Data normalisation was developed in the 1970s. It also gave rise to a standardised method of querying data, Structured Query Language (SQL). What is the difference between the queries created in Access and using SQL? What is the process of normalisation and how is it used?

EXAM TIP

The exam for this topic is a set of practical tasks based around a given scenario. You will be provided with an Access database and asked to improve certain existing features and add new ones such as a data entry form. You may also be asked to identify errors in the existing data or structure.

CHECK POINT

1. When naming database fields, what is considered good practice?
2. What name is given to a key value from a table at the 'one' end of relationship that is inserted into a record at the 'many' end?
3. What does Referential Integrity do?
4. How can the Boolean datatype be used?
5. To record the quantity of items a customer has purchased what datatype should you use?
6. What is the name given to a key that is made up from more than one field?

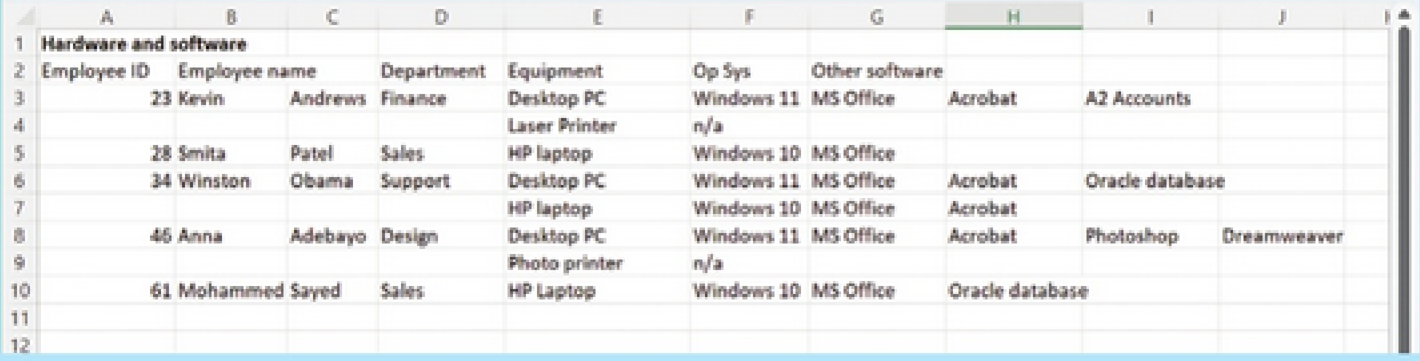
EXAM PRACTICE

1. Portia is the IT manager of a small company. She keeps a spreadsheet to show what hardware and software each employee has. Each employee can have one or more devices (such as a laptop, desktop computer, printer etc.) and each device can have any number of software applications installed on it.

   Portia is finding it difficult to keep track of the hardware and software using a spreadsheet and wants a solution to make it easier to manage this data.

   She wants a solution that will allow her to manage the system more effectively.

   An extract of the spreadsheet is shown:



	A	B	C	D	E	F	G	H	I	J
1	**Hardware and software**									
2	Employee ID	Employee name		Department	Equipment	Op Sys	Other software			
3	23	Kevin	Andrews	Finance	Desktop PC	Windows 11	MS Office	Acrobat	A2 Accounts	
4					Laser Printer	n/a				
5	28	Smita	Patel	Sales	HP laptop	Windows 10	MS Office			
6	34	Winston	Obama	Support	Desktop PC	Windows 11	MS Office	Acrobat	Oracle database	
7					HP laptop	Windows 10	MS Office	Acrobat		
8	46	Anna	Adebayo	Design	Desktop PC	Windows 11	MS Office	Acrobat	Photoshop	Dreamweaver
9					Photo printer	n/a				
10	61	Mohammed	Sayed	Sales	HP Laptop	Windows 10	MS Office	Oracle database		
11										
12										

(a) Create an efficient data structure to store this data in a way that minimises data duplication, uses correct data types and primary keys, and enforces referential integrity. Add any additional fields that may be required.

Produce a screen print that clearly shows the relationships, including table names, field names and primary keys.

Produce screen prints showing the field names and data types in the tables you have created. [6 marks]

(b) Portia wants a user-friendly form to store the details of new employees.

The form must:

- open ready for data entry
- generate a new EmployeeID that is the highest existing number plus 1
- allow a department from a choice of Sales, Finance, Support, Design and Management only
- not allow a record to be saved without an employee family name
- have a save button that saves a record, displays a confirmation message and clears the form for a new record.

Provide screen prints of your form in Design view and Form view. [8 marks]

TOPIC 19 DATABASE SOLUTIONS

We have already covered how relational databases are designed and the table structures created. Now we need to extend that understanding by looking at how database applications can be developed and how advanced features, such as data import and macros, can be used. We also need to consider how a database can be tested to ensure it works as it should and how we can evaluate the various features of a database application.

TOPIC 19 DATABASE SOLUTIONS

LEARNING OBJECTIVES

By the end of this chapter you should be able to understand:

- how to use database software to create queries, and reports
- how to import and export data to and from a database
- how to retrieve data and use it to produce meaningful information in a format that is appropriate for the user
- how to create useful user interfaces for database applications and evaluate their use
- how to write macros to automate database functions
- how to test database features
- how to evaluate and enhance a database application.

GETTING STARTED

Understanding the basics of relational database design is at the heart of this topic. Can you remember from Topic 18 what are the three types of relationships that can exist between database entities? Can you think of a real-life example of the most common type of relationship? What type of relationship cannot be represented in a relational database? How is this resolved? Entity relationship diagrams (ERD) are used to model database relationships. How are these diagrams drawn and how are different types of relationship represented?

LANGUAGES
HELD
ORGANIZING
HIERARCHICAL
NEEDED
MANIPULATED
USED
CONVENTIONAL
POSITIONS
SEE
STORAGE
CONTENT
EVEN
EXPLICIT
DIFFERENCES
APPROACHED
STATES
SPANNING
LARGE
TWO
MANY
COLUMN
KIND
CONTEXT
OBJECT
COMPUTER
DATABASE
SYSTEM
ACCESS
QUERY
INFORMATION
ENTIRE
WELL
RELATIONSHIPS
MODEL
STRUCTURED
APPLICATION
USERS
CATEGORIZED
LANGUAGE
APPLICATIONS
ISCUSSION

▶ Are you familiar with database terminology?

USING DATABASE SOFTWARE

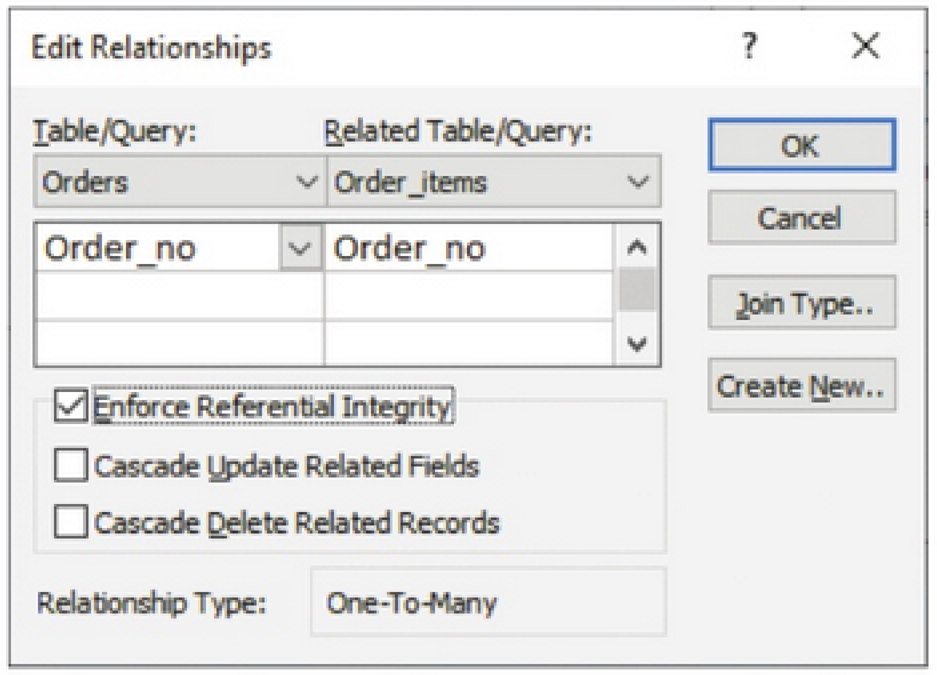
UPDATE, MODIFY, INSERT AND DELETE DATA

You have already seen in Topic 18 how to design a relational database and create the relationships needed to structure the data. You also looked at how forms can be developed to provide an easy-to-use method for users to input data. We now need to develop that knowledge further by looking at how facilities can be provided to find, amend and delete data and how to use reports to present data.

All of these tasks can be carried out in Access using the Datasheet view, but this interface, while very flexible, is not very user friendly and is not well suited for creating an end-user database application.

Deleting records in a relational database needs to be considered carefully. This is particularly the case if the record is in a table that is at the 'one' end of a one-to-many relationship. Consider the Orders database that you looked at in Topic 18. The Orders table is related to the Order_items table. One order can have many Order_items. You can of course delete a record in the Orders table, but what about any Order_items records that are related? Unless you delete the related records they will become 'lost', in that they won't have a corresponding Order record.

When you set referential integrity for relationships you may have noticed that there are two other options in the dialogue, as shown in Figure 19.1.


Edit Relationships

Table/Query: Orders

Related Table/Query: Order_items

Order_no | Order_no

☑ Enforce Referential Integrity

☐ Cascade Update Related Fields

☐ Cascade Delete Related Records

Relationship Type: One-To-Many

▲ Figure 19.1 The edit relationships dialogue

Cascade update related records. If you select this option and you make a change (update) to the primary key value of a table that is at the one end of a one-to-many relationship, then Access will also update all the foreign key values in related records in the table at the 'many' end.

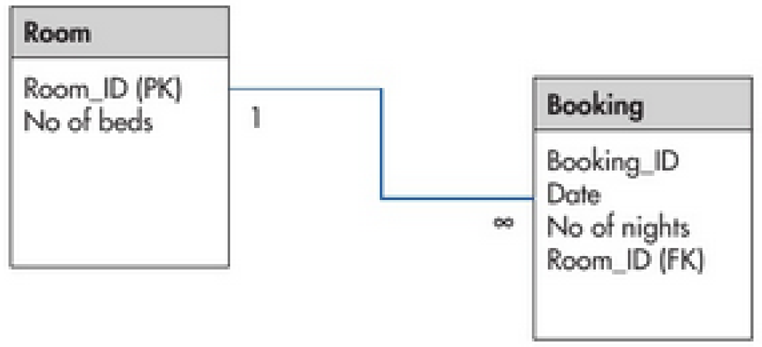
Cascade update related records. With this option selected, if you delete a record at the 'one' end of a one-to-many relationship it will also delete related records in the table at the 'many' end.

Which option should be selected will depend on the application and may need some careful thought. What should happen, for example, if a customer cancels an order? In many real-world database applications, records that need to be deleted are placed in a 'deleted' table so they can still be accessed if needed.

CREATING RELATIONAL DATABASE STRUCTURES

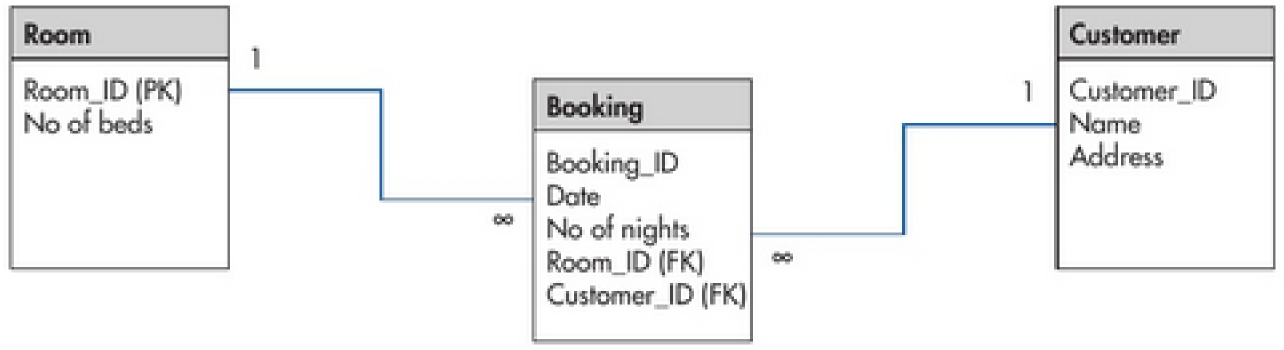
Creating relational database structures using Access relationship tools has been covered in Topic 18.

Suppose a small hotel needed a database to track the booking of its rooms. Hotel customers book rooms. This actually defines three 'entities' in the application. The physical entities of rooms and customers and the 'virtual' entity of a booking. Thinking of it in its simplest form, a booking can only be for one room. Over time, a room can have many bookings. Therefore a rooms table will have a one-to-many relationship with a bookings table. Placing the primary key of the rooms table in a booking record will identify which room a booking is for. Figure 19.2 shows an ERD for this relationship.



▲ Figure 19.2 The relationship between rooms and bookings

Each customer can make many bookings so the customer has a one-to-many relationship with the booking. This is shown in Figure 19.3.



▲ Figure 19.3 The customer table added to the ERD

SKILLS

PROBLEM SOLVING

ACTIVITY 1

In reality, a hotel database would be much more complex than this. For example, a customer might want to make a single booking for more than one room. This would create a many-to-many relationship between booking and rooms. How could this be resolved?

Have a go at drawing an ERD for the relationship between customers, bookings and rooms.

KEY TERM

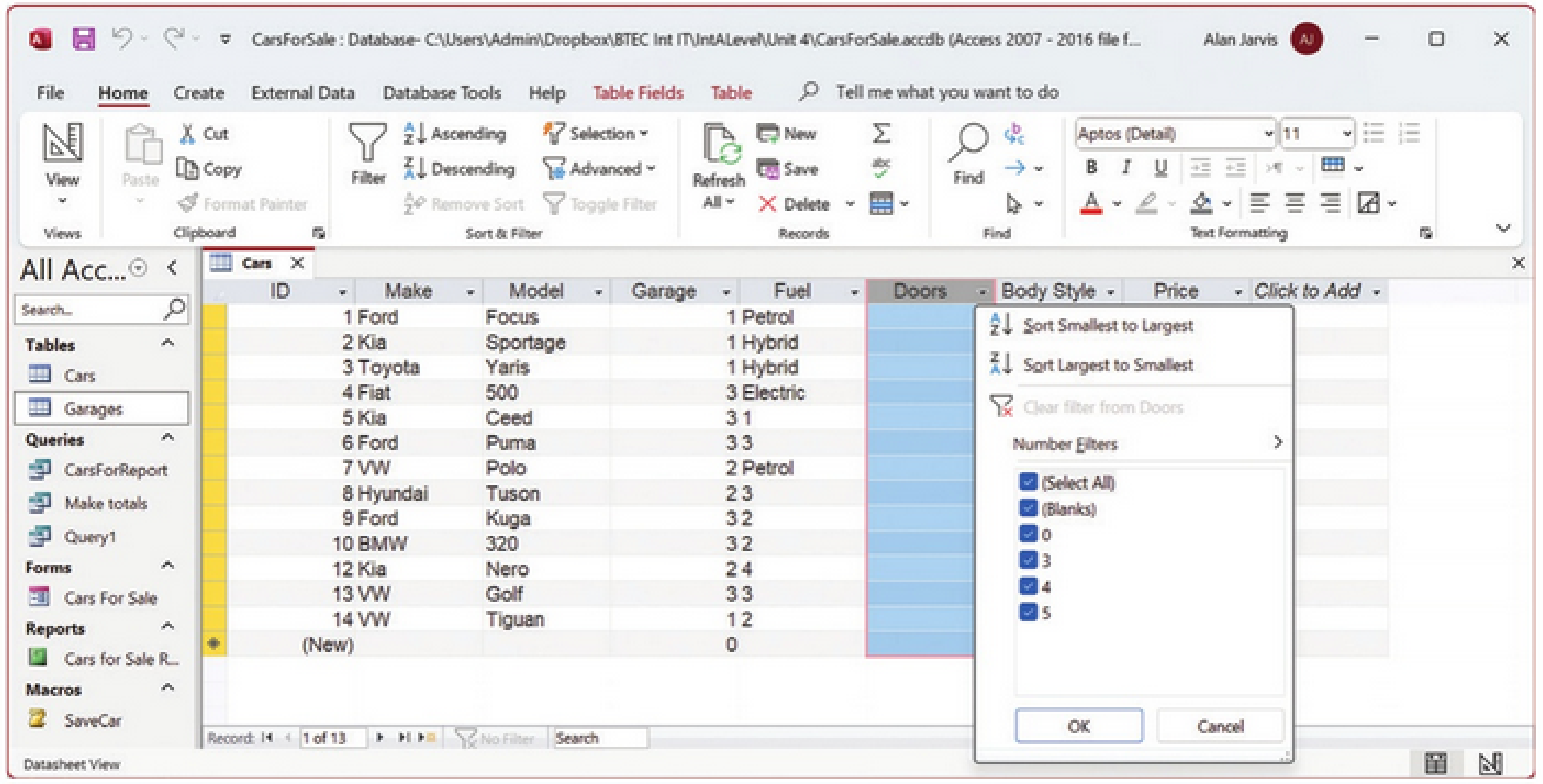
criteria: a definition of how a query should select data to be displayed

RETRIEVING DATA

You have covered data entry but not retrieving data from the database. Most applications will have a requirement to retrieve a single record – a customer call centre, for example, will need to access the customer's data when they call, and also retrieve all the records that meet given **criteria**. It is also common to have a requirement to find a set of records that match a particular criterion, such as all those accounts that have overdue payment.

Filters

These provide the simplest method of selecting records that match certain criteria. To filter records displayed in Datasheet view you can use the drop-down arrows that appear in the field names across the top of the datasheet. Clicking on the arrow will show a menu that you can choose to filter options from. Figure 19.4 shows the filter drop-down menu.



CarsForSale : Database- C:\Users\Admin\Dropbox\BTEC Int IT\IntALevel\Unit 4\CarsForSale.accdb (Access 2007 - 2016 file f...
Alan Jarvis
File
Home
Create
External Data
Database Tools
Help
Table Fields
Table
Tell me what you want to do
View
Paste
Cut
Copy
Format Painter
Filter
Ascending
Descending
Remove Sort
Selection
Advanced
Toggle Filter
Refresh All
New
Save
Delete
Find
Views
Clipboard
Sort & Filter
Records
Find
Text Formatting
All Acc...
Search...
Tables
Cars
Garages
Queries
CarsForReport
Make totals
Query1
Forms
Cars For Sale
Reports
Cars for Sale R...
Macros
SaveCar
Sort Smallest to Largest
Sort Largest to Smallest
Clear filter from Doors
Number Filters
(Select All)
(Blanks)
0
3
4
5
OK
Cancel
Record: 1 of 13
No Filter
Search
Datasheet View

ID	Make	Model	Garage	Fuel	Doors	Body Style	Price	Click to Add
1	Ford	Focus	1	Petrol				
2	Kia	Sportage	1	Hybrid				
3	Toyota	Yaris	1	Hybrid				
4	Fiat	500	3	Electric				
5	Kia	Ceed	3	1				
6	Ford	Puma	3	3				
7	VW	Polo	2	Petrol				
8	Hyundai	Tuson	2	3				
9	Ford	Kuga	3	2				
10	BMW	320	3	2				
12	Kia	Nero	2	4				
13	VW	Golf	3	3				
14	VW	Tiguan	1	2				
(New)			0					

▲ Figure 19.4 The filter drop-down menu

The filter options are:

Sort. To sort the data by the chosen field, use the ascending (A–Z) or descending (Z–A) buttons.

Number filters. Clicking on this option will pop another menu out from where you can choose comparison filters such as 'equal to', 'greater than', 'less than', etc. When you choose the filter you want it will then pop a dialogue onto the screen asking you the value that you want the comparison made with.

Individual values. All the different individual values in the field are listed here, and you can click in the check boxes next to the value(s) you want to filter.

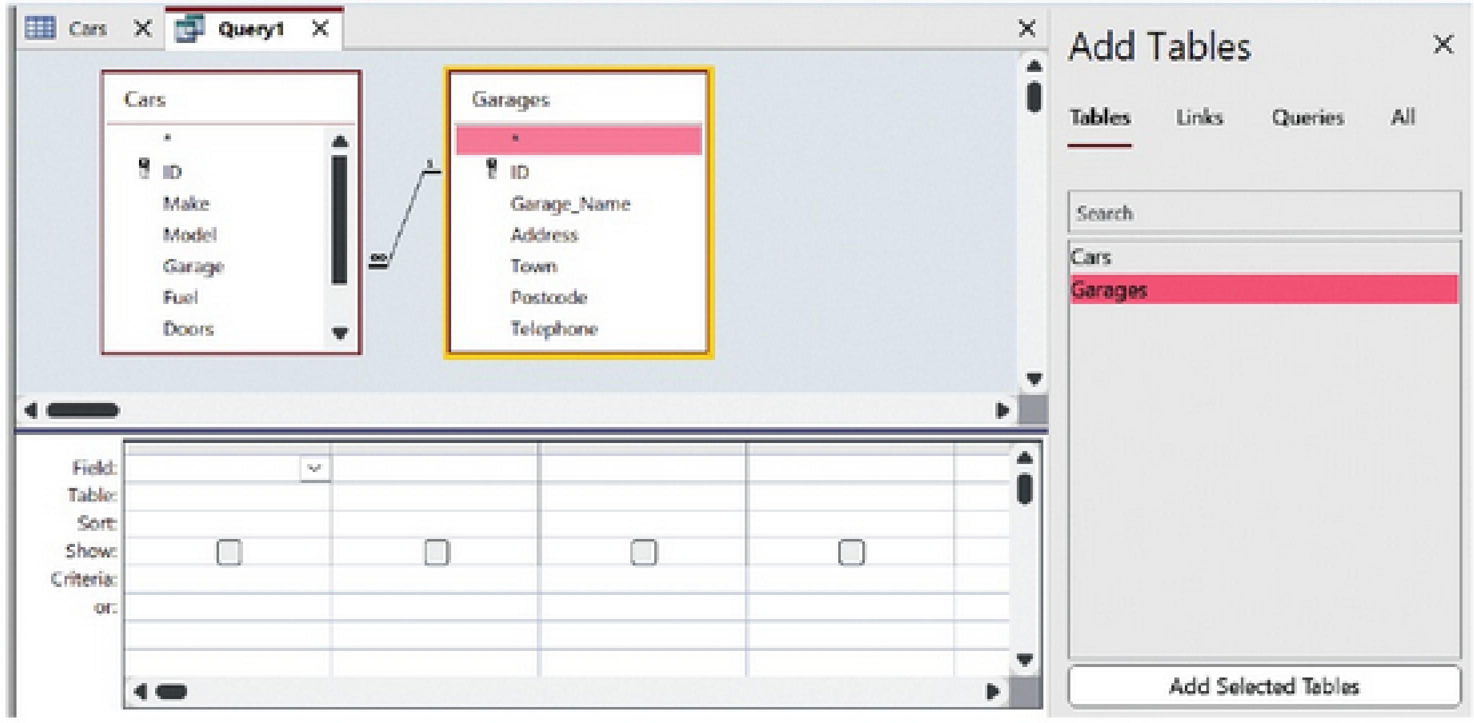
When the filter is applied, only the matching records are displayed in the datasheet, and a filter symbol appears in the field name header row of the field where the filter has been applied. You can apply additional filters to the filtered datasheet using other fields.

Queries

Queries are more sophisticated than filters and can be saved so you can run them whenever needed. Queries can also be built into a database application to provide end-user facilities without the user needing to understand how to create a **query**.

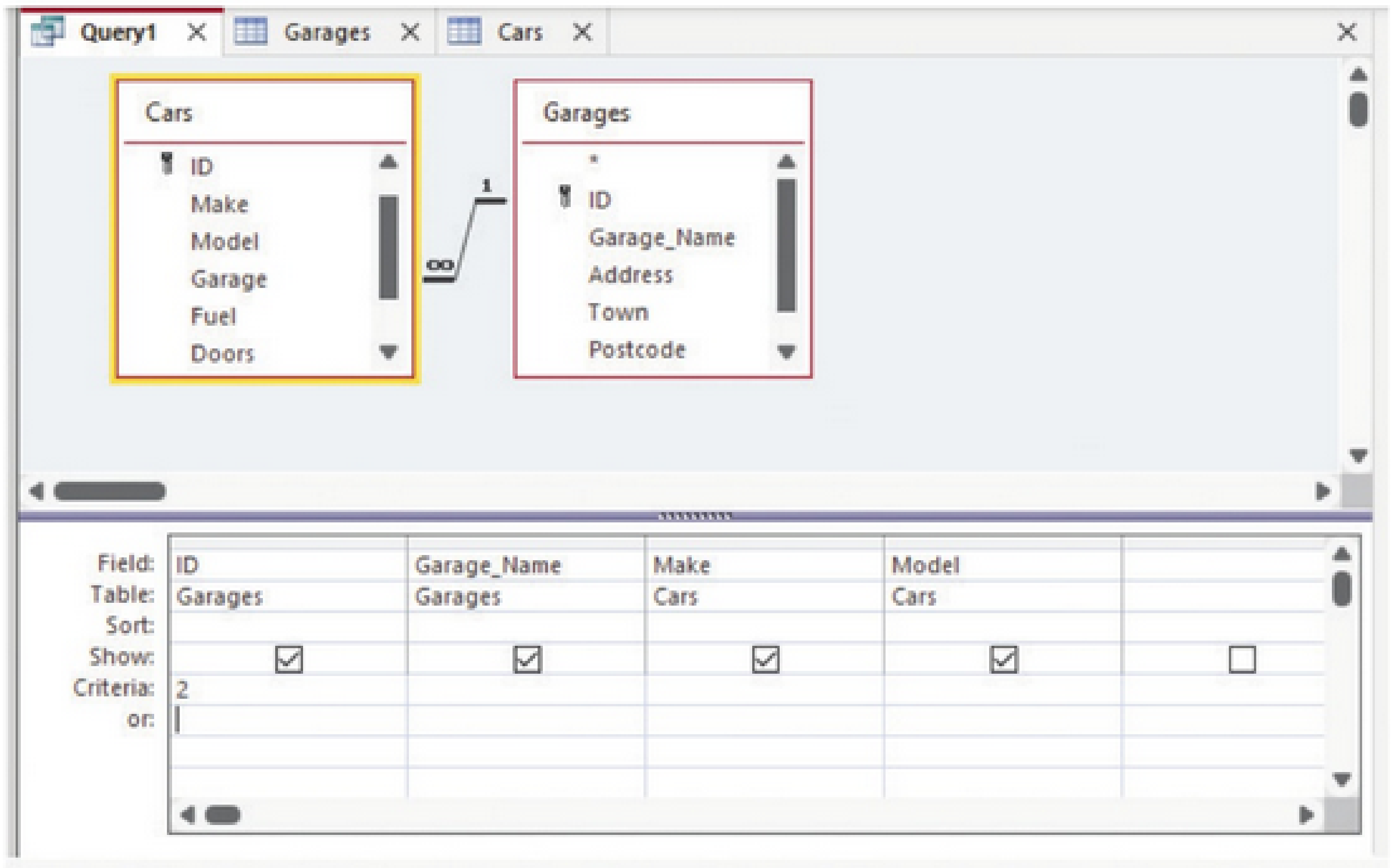
To create a query, go to the Create menu and choose Query Design. This will open the query grid. Make sure you have the Add Table pane open; if not, go to the Query Design menu and choose Add Tables to display it. In this example we will use the Cars for Sale database that you looked at in Topic 18. Both the Garage and the Cars tables have been added to the query grid, which now looks like Figure 19.5.

Suppose you just wanted to display a list of those cars from a particular garage. To do this, you would need to filter cars displayed based on the Garage ID, for example, cars from Southgate Garage, which has the ID of 2. Drag the Garage ID down from the table in the top half of the grid into the first row and first column of the query grid. Then, so you can check we have the right garage, drag the Garage-Name down into the second column of the query grid. If you wanted to see the make and model of the cars, you could drag those two fields down from the Cars table into the 3rd and 4th column. Then you would need to set the criteria. The Garage ID is 2, so type a 2 in the criteria row of the first column. See Figure 19.6.



Cars
Query1
Cars
*
ID
Make
Model
Garage
Fuel
Doors
Garages
*
ID
Garage_Name
Address
Town
Postcode
Telephone
1
∞
Field:
Table:
Sort:
Show:
Criteria:
or:
Add Tables
Tables
Links
Queries
All
Search
Cars
Garages
Add Selected Tables

▶ Figure 19.5 The query grid with the Garage and Cars tables added



Query1
Garages
Cars
Cars
ID
Make
Model
Garage
Fuel
Doors
Garages
*
ID
Garage_Name
Address
Town
Postcode
1
∞

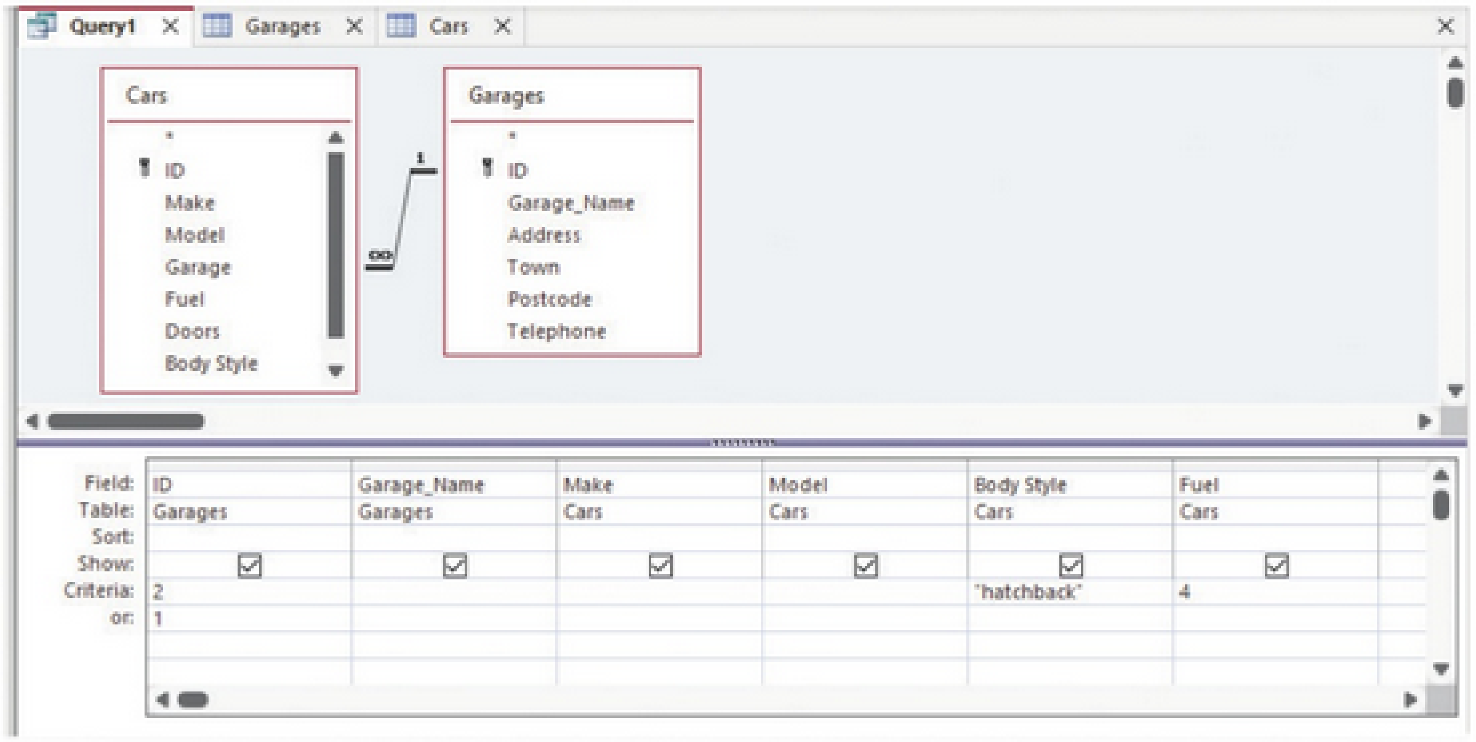
Field:	ID	Garage_Name	Make	Model	
Table:	Garages	Garages	Cars	Cars	
Sort:					
Show:	☑	☑	☑	☑	☐
Criteria:	2				
or:					

▶ Figure 19.6 The completed query grid

To run the query and display the matching records, click the Run icon in the Query Design menu and you will see the matching records displayed in Datasheet view. To return to Query Design, click the Design icon on the left side of the Home menu.

The row under the Criteria row is labelled 'or'; you can add another criteria in that row to select records that meet either the first criteria or the one in the 'or' row. So, you could add another Garage ID code to get cars at either garage. The rows below that, which are not labelled, can be used to add further criteria.

You can add multiple criteria if required. Suppose you wanted an electric car at the Southgate Garage with a hatchback body style. You would need to add the Fuel and Body style fields and add criteria for each. Note that the fuel is shown using a number, with 4 representing electric (this is because, in Topic 18, you entered this data using a combo box). Figure 19.7 shows this query.



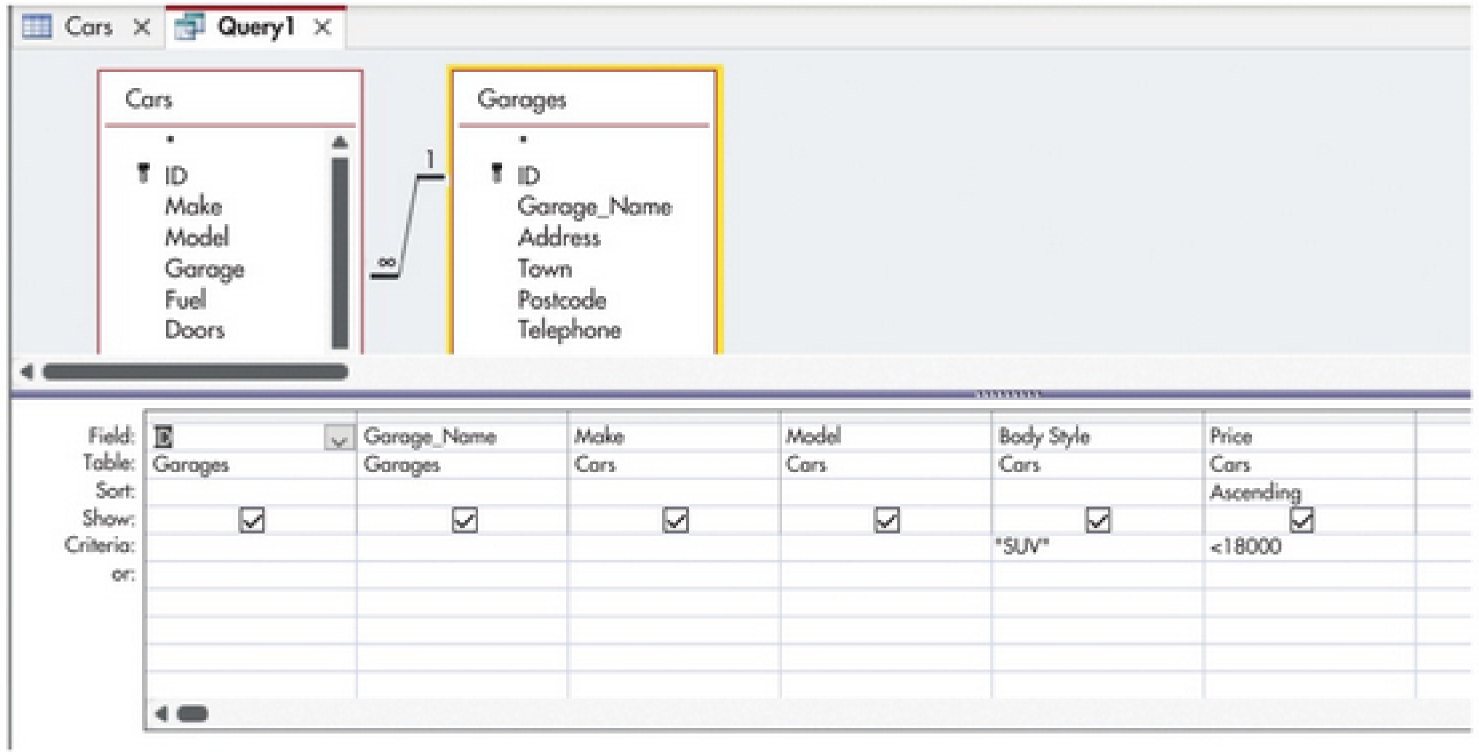
Field:	ID	Garage_Name	Make	Model	Body Style	Fuel
Table:	Garages	Garages	Cars	Cars	Cars	Cars
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:	2				"hatchback"	4
or:	1					

▶ Figure 19.7 Query for electric hatchback cars at the Southgate garage

Click the Save button in the top left of the Window to save the query so you can run it whenever needed.

In the query grid there is a row labelled Sort. Clicking within this row under one of the fields allows you to sort that field in either ascending or descending order. There is also a 'Show' row with a check box for each field. Uncheck the box and the field will not be displayed when the query runs, although any criteria set using that field will still apply.

With a numeric field you can add criteria that use comparison symbols such as greater than (>) or less than (<). So if you wanted to find cars with SUV body style under £18,000, sorted by price, the query grid should look like Figure 19.8.



Field:	ID	Garage_Name	Make	Model	Body Style	Price
Table:	Garages	Garages	Cars	Cars	Cars	Cars
Sort:						Ascending
Show:	☑	☑	☑	☑	☑	☑
Criteria:					"SUV"	<18000
or:						

▶ Figure 19.8 Query to find SUV body style cars under £18,000, sorted by price

Note that if you put the second criteria (<18000) in the 'or' row rather than in the 'Criteria' row, then the query would find cars that have a SUV body style OR are below £18000. You can also use a range in a criterion, so if you entered 'Between 18000 and 20000' in the criteria cell then you would have cars with a price between £18,000 and £20,000 listed.

In the query criteria you can also use 'wildcards' for text values. Suppose you added a field to this database that records the car's colour. There might be values in that field such as 'light blue', 'sky blue' and 'blue/black' (for a two tone colour scheme). To find all the cars with any type of blue paintwork you would enter a criterion of '*blue*'. The asterisk (*) represents any other characters before or after the word 'blue'.

You can also use the ? symbol to match just a single character. So 'a?c' would find 'abc' and 'azc', but not 'abd'. This could be useful to search for cars with number plates that contain a certain string, which might identify where and/or when a car was first registered.

You also can use dates with query criteria, some examples are:

Date(). This represents the current date, so you can use this in a query criterion to find records that have a field set to the current date.

Comparison operators. You can use '<' or '>' to find any records with a date before or after the one entered.

Between. You can use between with two dates to find any records with dates between the given dates, for example between 1/1/24 and 20/1/24. Access will automatically add # symbols before and after the dates.

Date Part. You can find records that match a given month by using the DatePart function. For example DatePart("m",[order_date])=2 will find any order_date records in the month of February; the "m" tells the function to look for months and the '2' sets it to February. This criterion: DatePart("yyyy",[order_date])=2024 would find all the order_dates in 2024.

SKILLS

PROBLEM SOLVING

ACTIVITY 2

1. Consider a database system used by a library to keep track of its books. Books have a title, a category (e.g. science, history, IT, fiction, etc.) and an author. (Assume for this example that every book only has a single author.) A library also has people who borrow books (borrowers). Each borrower can borrow several books. The table shows a data extract from the library system.

Book ID	Book title	Author ID	Author name	Category ID	Category name	Borrower ID	Borrower name	Book on loan
1	History of Asia	2	Ahmed Patel	2	History			No
2	Basic Physics	3	Nisha Kumar	3	Science	4	Ercan Akhtar	Yes
3	Introduction to Access	5	Smita Jamdar	1	Computing			No
4	Sea fishing	6	Clive West	4	Hobbies	2	Nissi Ampoma	Yes
5	The Roman Empire	2	Ahmed Patel	2	History			No
6	Plant Biology	1	Ethan Jones	3	Science	1	Bob Ragoo	Yes
7	Web page design	5	Smita Jamdar	1	Computing	3	Mina Raj	Yes
8	Gardening	6	Clive West	4	Hobbies			No
9	World Wars	2	Ahmed Patel	2	History	1	Bob Ragoo	Yes
10	Python programming	5	Smita Jamdar	1	Computing	3	Mina Raj	Yes

Create an ERD and database tables for this data. Then enter the data, adding some further records if you can.

Create and test the following queries using the Query Design menu:

a. Find all the books written by Ahmed Patel.

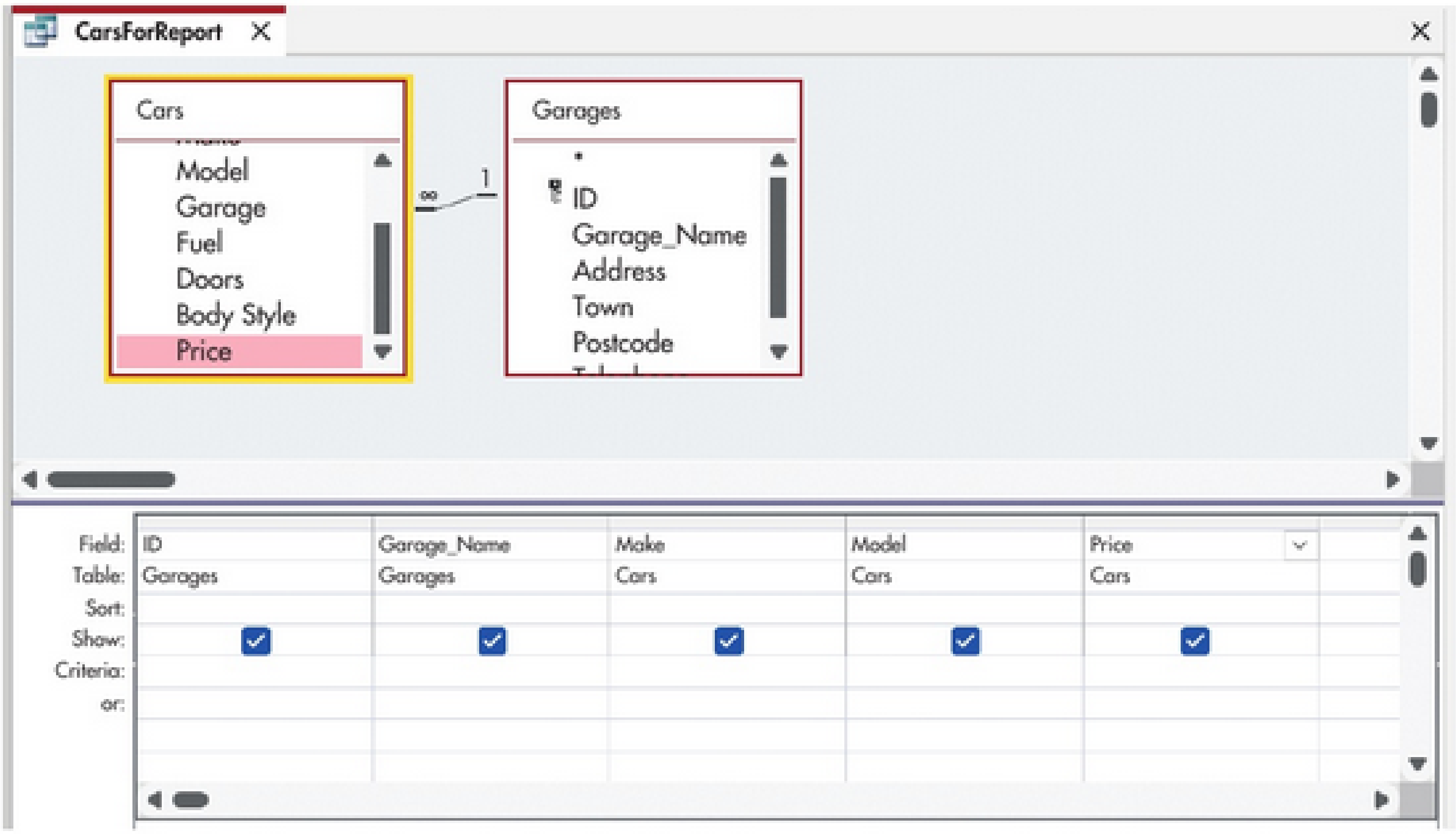
b. How many books does Mina Raj have on loan?

c. Find all the history books that are not on loan.

REPORTS

Reports are designed for printing rather than being interactive like forms. Reports therefore focus on laying out and formatting data on a page, grouping data and adding totals, averages and other calculations. Reports can take data from a table or a query, and you can bring together and filter data from several tables with a query then format it for printing with a report.

Let's look at an example of creating a report for the Car database you have been using. Suppose a report was needed to list all the cars by garage, showing the total value of cars for sale at each garage and across all the garages. Because this report will need to bring together data from two tables (the Cars table and the Garage table), you will need to use a query to do that. You only need to include the Garage ID and Name fields from the Garage table and the Make, Model and Price fields from the Cars table. The query grid for this is shown in Figure 19.9. Since you want all the cars from all the garages, no criteria are required. The query has been saved and named 'CarsForReport'.



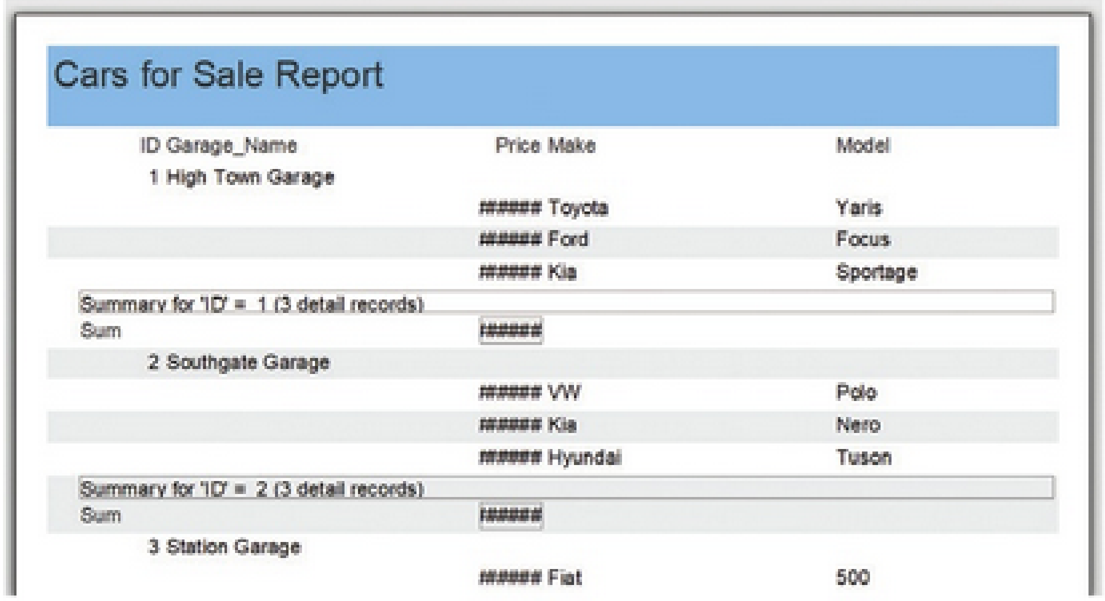
Field:	ID	Garage_Name	Make	Model	Price
Table:	Garages	Garages	Cars	Cars	Cars
Sort:					
Show:	☑	☑	☑	☑	☑
Criteria:					
or:					

▲ Figure 19.9 The query grid used to provide data for the report

The next step is to create the report using the Report Wizard. Choose the Create menu then the Report Wizard icon on the right side of the toolbar. This will open the Report Wizard. The steps of the Wizard are as follows:

1. Make sure the CarsForReport is selected as the table/query and add all the fields from the query to the 'Selected fields' box, then click Next.
2. The next page of the Wizard will show how the data will be grouped, with the garage as the top level with cars from each garage shown under the Garage, which is exactly what you want. Click Next.
3. This page asks if you want to add any further groupings, which you don't, so click Next.
4. Here you are asked if you want to sort data in any way. The car prices could be sorted, so choose Price and leave this set as Ascending. Then click the Summary Options button, which opens a pop-up box. Here the summary option of 'sum' is selected so the car prices will be totalled up. Click OK on this box. Then click Next on the Wizard page.
5. This page sets layout options, so there is no need to change any of these. Click Next.
6. This is the final page of the Wizard. The title of the report can be changed to something a bit more descriptive, such as 'Cars for Sale Report'. Click Finish.

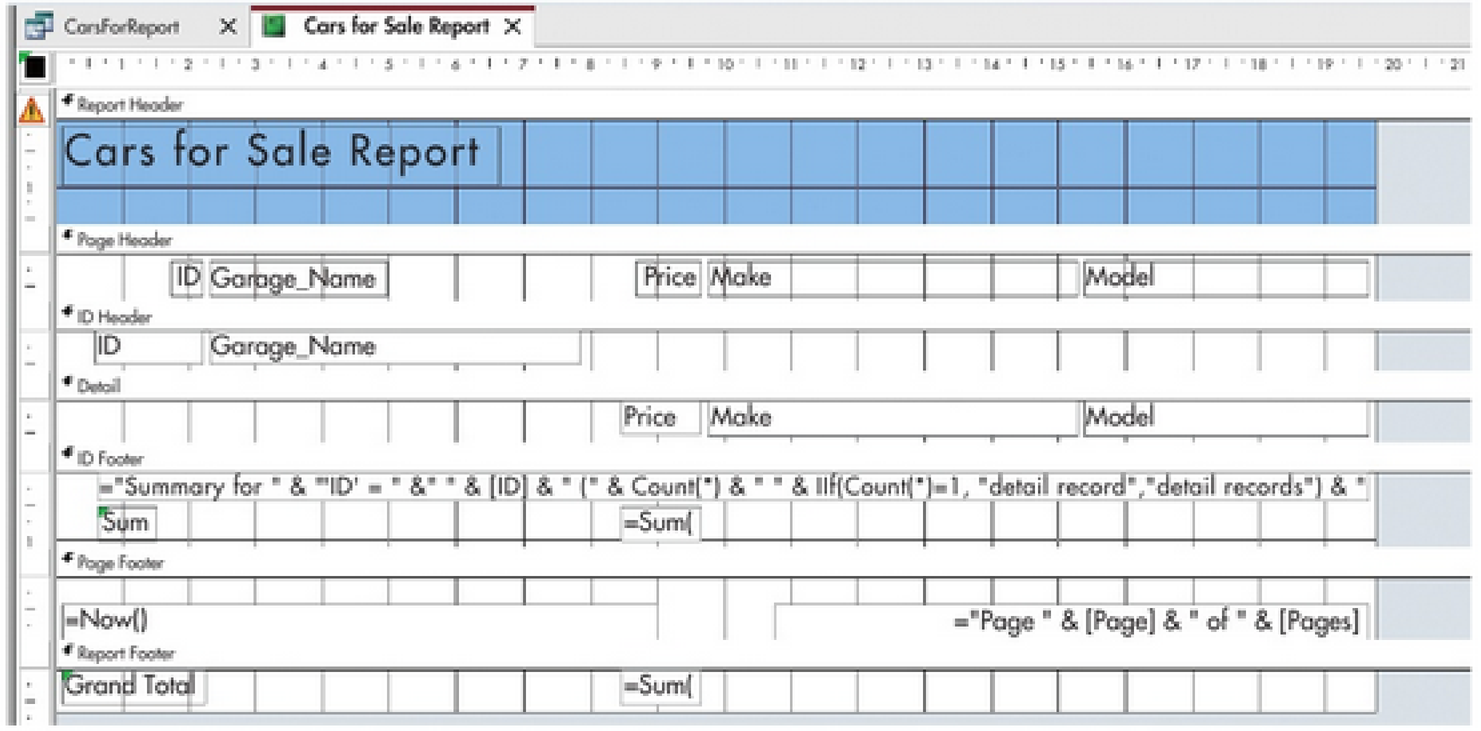
The report will then be shown in preview mode. Figure 19.10 shows an example of the report. As you can see, the report needs some adjustments.



```
Cars for Sale Report
    ID Garage_Name              Price Make              Model
     1 High Town Garage
                               ###### Toyota            Yaris
                               ###### Ford              Focus
                               ###### Kia               Sportage
  Summary for 'ID' =  1 (3 detail records)
  Sum                          ######
     2 Southgate Garage
                               ###### VW                Polo
                               ###### Kia               Nero
                               ###### Hyundai           Tuson
  Summary for 'ID' =  2 (3 detail records)
  Sum                          ######
     3 Station Garage
                               ###### Fiat              500
```

▲ Figure 19.10 The report created by the wizard

The main problem with the report created in this example is that the prices are not displayed because the field is too small (shown by the # symbol). Click the Close Print Preview button on the right and the report will be displayed in Design view (Figure 19.11).

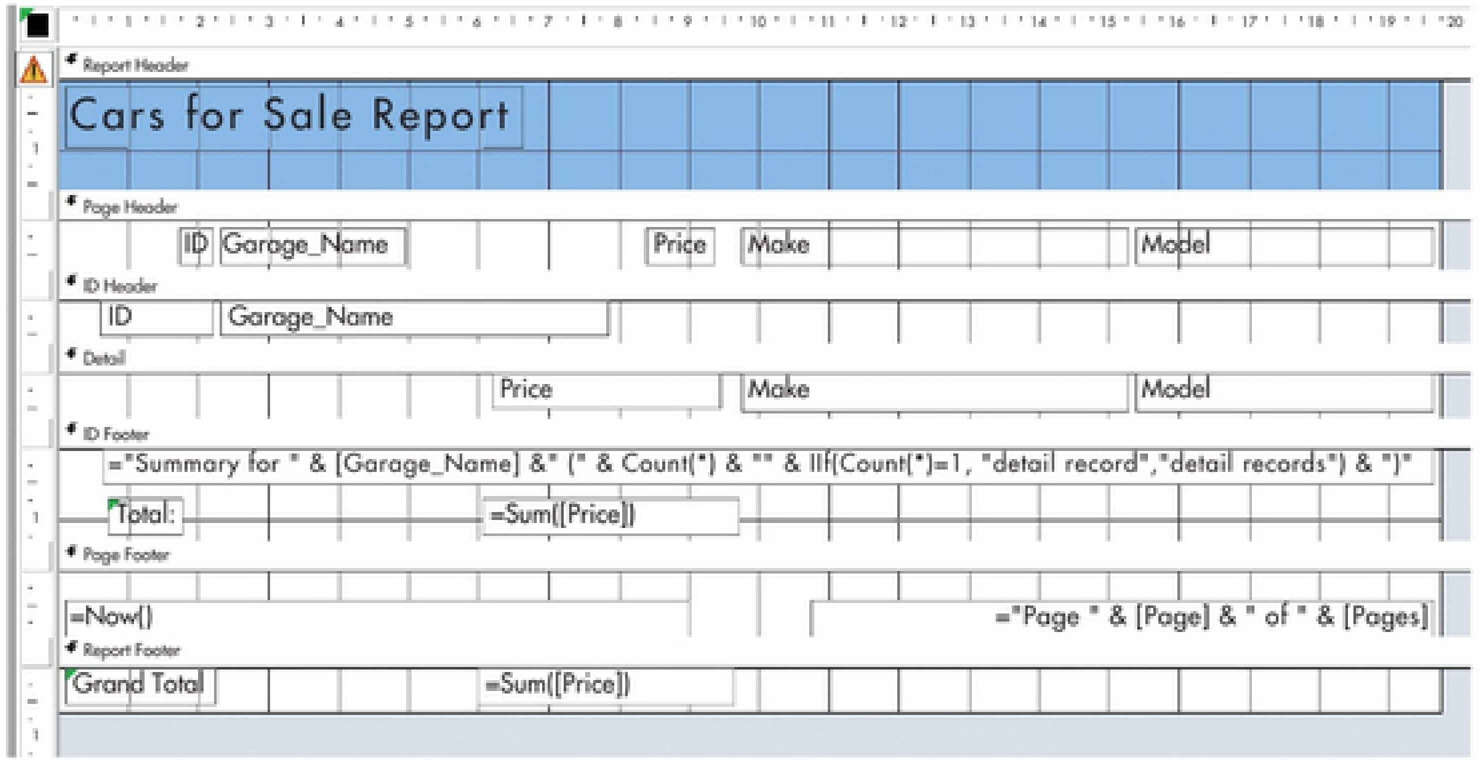


```
Cars for Sale Report
ID Garage_Name          Price Make          Model
ID      Garage_Name
                        Price Make          Model
="Summary for " & "'ID' = " &" " & [ID] & " (" & Count(*) & " " & IIf(Count(*)=1, "detail record","detail records") & "
Sum                     =Sum(
=Now()                              ="Page " & [Page] & " of " & [Pages]
Grand Total             =Sum(
```

▲ Figure 19.11 The report in Design view

The title for the Garage totals is Garage_ID, but it would be better to use the name of the Garage.

Figure 19.12 shows the edited version of the design for the report.


Report Header

Cars for Sale Report

Page Header

ID	Garage_Name	Price	Make	Model

ID Header

ID	Garage_Name

Detail

Price	Make	Model

ID Footer

```
="Summary for " & [Garage_Name] &" (" & Count(*) & "" & IIf(Count(*)=1, "detail record","detail records") & ")"
```

Total: =Sum([Price])

Page Footer

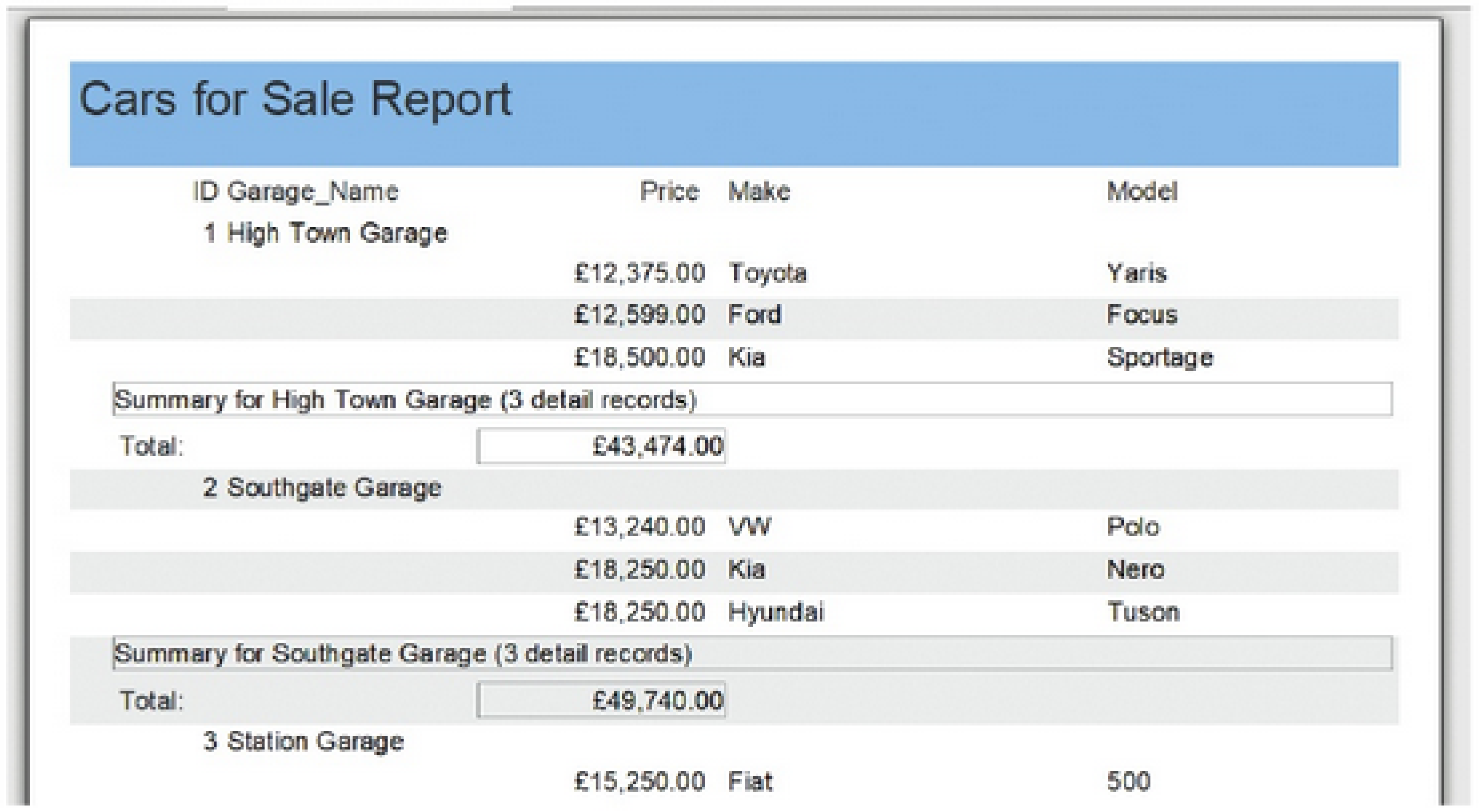
=Now()   ="Page " & [Page] & " of " & [Pages]

Report Footer

Grand Total =Sum([Price])

▲ Figure 19.12 The edited Design view for the report

Figure 19.13 shows the improved version of the report in print preview mode.


Cars for Sale Report

ID Garage_Name	Price	Make	Model
1 High Town Garage			
	£12,375.00	Toyota	Yaris
	£12,599.00	Ford	Focus
	£18,500.00	Kia	Sportage
Summary for High Town Garage (3 detail records)			
Total:	£43,474.00		
2 Southgate Garage			
	£13,240.00	VW	Polo
	£18,250.00	Kia	Nero
	£18,250.00	Hyundai	Tuson
Summary for Southgate Garage (3 detail records)			
Total:	£49,740.00		
3 Station Garage			
	£15,250.00	Fiat	500

▲ Figure 19.13 The edited report

Editing reports in Design view is quite fiddly, and sometimes it can be hard to get the report laid out as you would like, but practice will improve your skills.

CALCULATED FIELDS IN REPORTS AND QUERIES

Queries provide database users with a method of answering questions about their data. This can help organisations understand financial and marketing issues such as the cost of providing a service, the profits they can make or the popularity of different products.

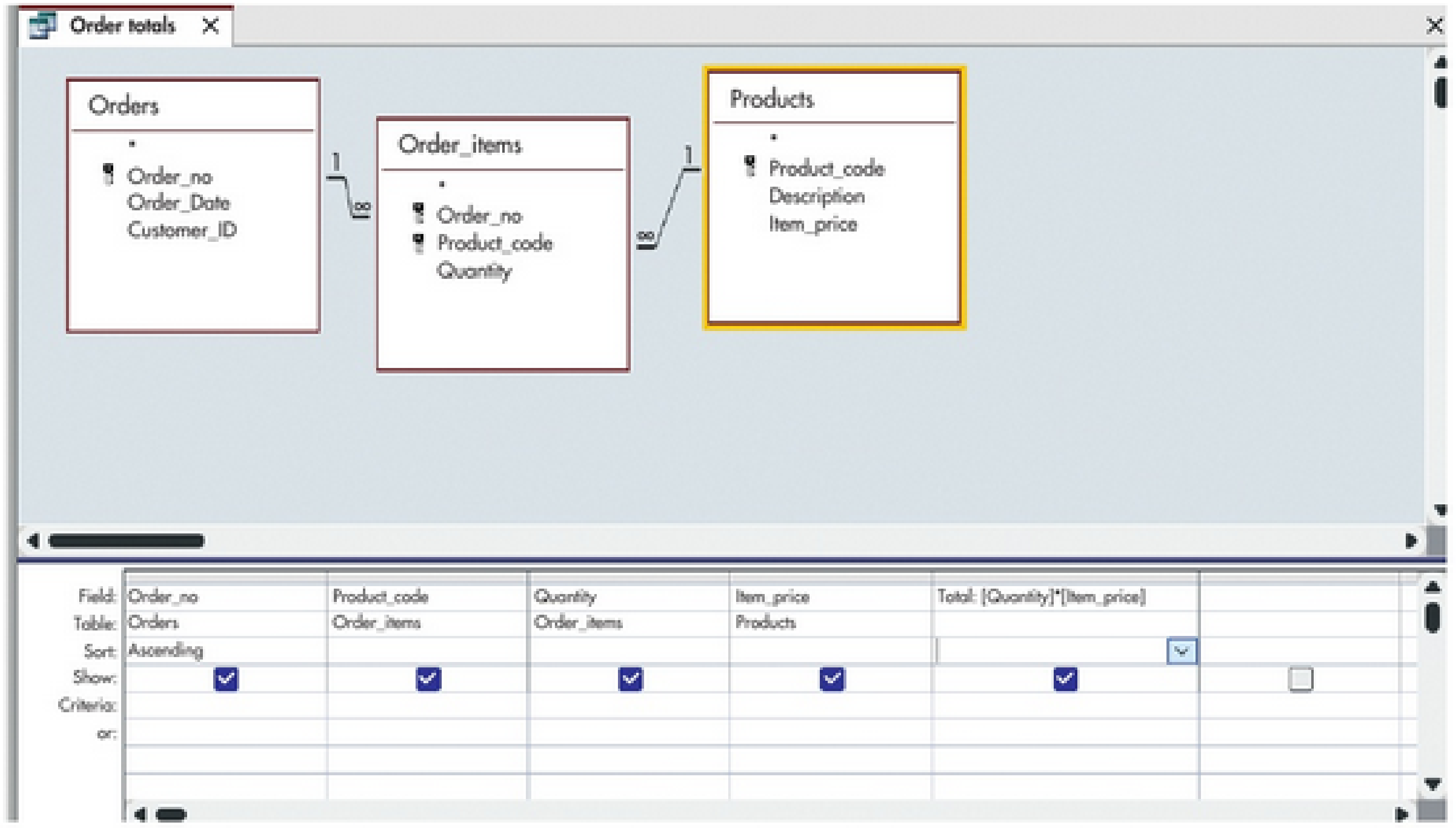
You have already seen that you can use the Report Wizard to automatically add totals to a report. You can also carry out a range of different calculations in a query.

Let's look at an example. You previously used the Orders database. This has an Orders table and a related Order_items table. In the Order_items table the product that has been ordered is shown by its product code. This provides a link to the Products table where the price of the item is listed. However, a customer can order any quantity of the item so the total price for the order item (quantity * price) needs to be calculated. This can be done by creating a new, calculated field in a query. The calculation is:

Total: [Quantity]*[Item_price]

Where Total is the name given to the calculated field.

Figure 19.14 shows the query grid with the calculated field included.

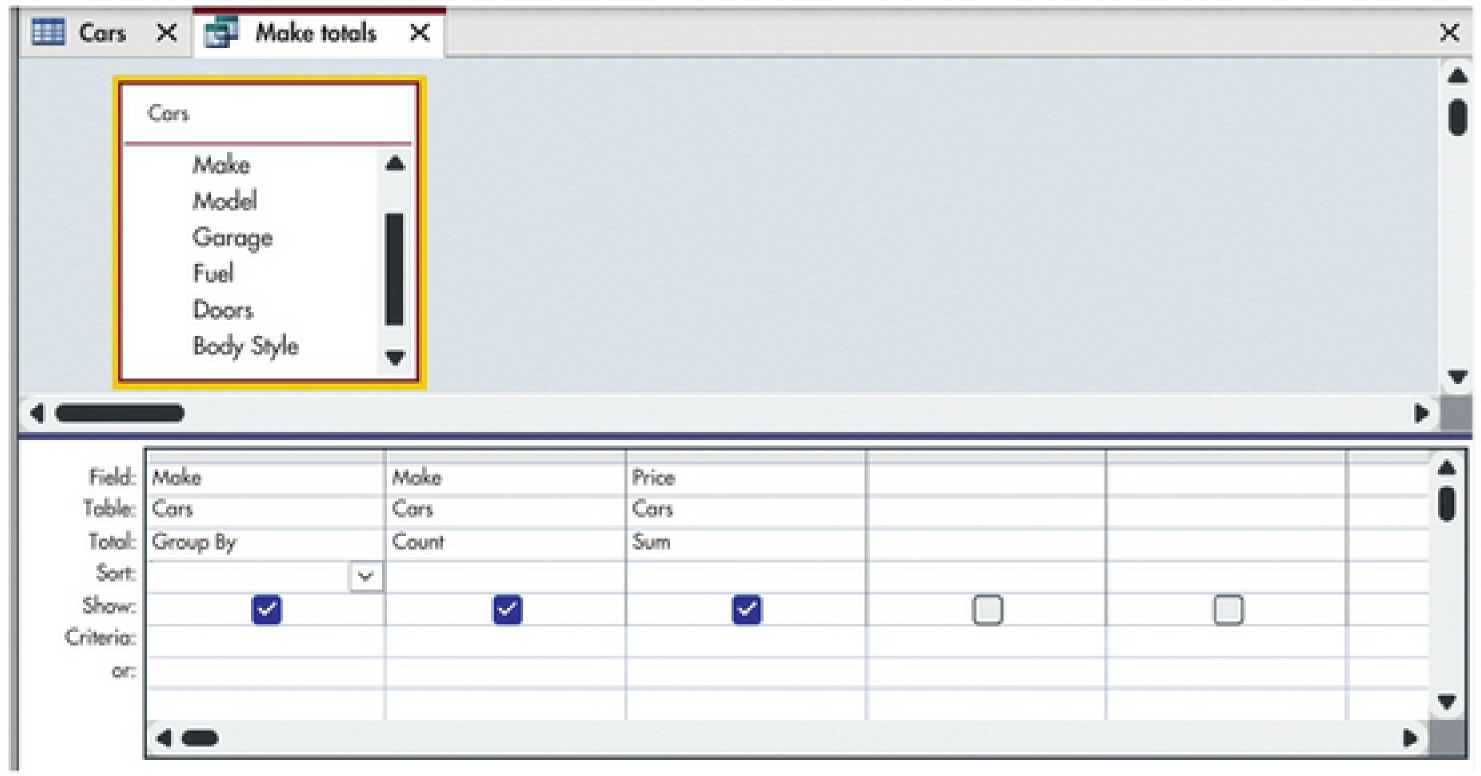


Field:	Order_no	Product_code	Quantity	Item_price	Total: [Quantity]*[Item_price]	
Table:	Orders	Order_items	Order_items	Products		
Sort:	Ascending					
Show:	☑	☑	☑	☑	☑	☐
Criteria:						
or:						

▲ Figure 19.14 The query grid showing a calculated field

When the query is run it will show the Total.

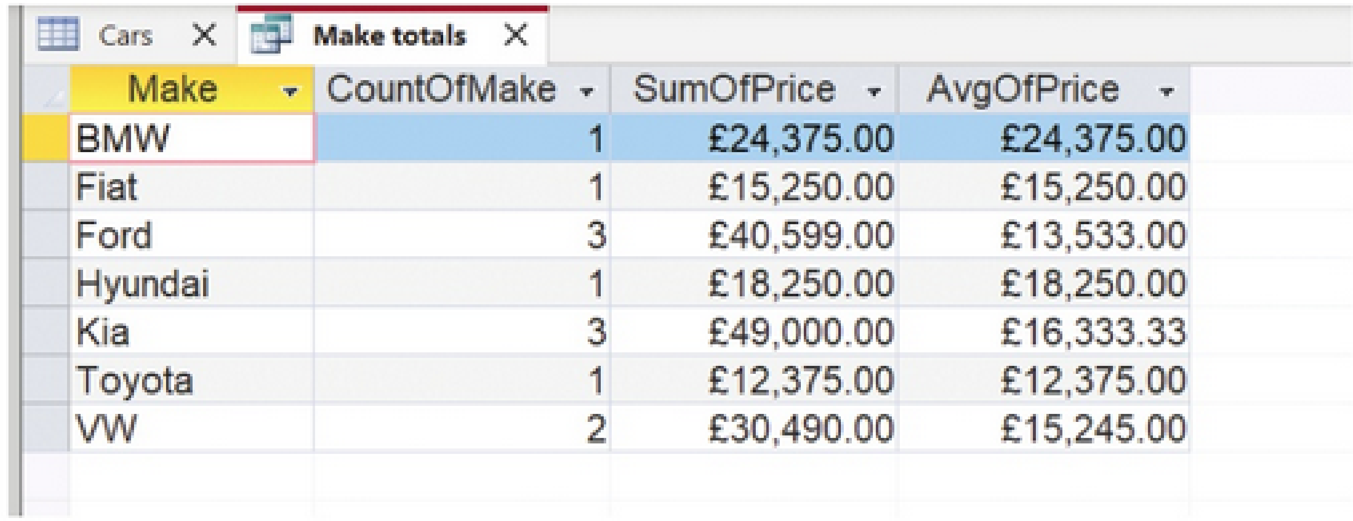
Another way you can use queries is to get the query to group the values in a certain field and then do a calculation based on the grouped values. As an example for this feature we will use the Cars for Sale database. Suppose the owner of the garages wanted to know how many cars were in stock by make (manufacturer). They would also like to know the total value of the cars in stock by make and the average price of cars by make. To add calculations such as total, count and average, you need to display the Total row in the query grid and then select the calculation you want to use. Figure 19.15 shows the use of Count and Sum calculations in the Totals row.



Field:	Make	Make	Price		
Table:	Cars	Cars	Cars		
Total:	Group By	Count	Sum		
Sort:					
Show:	☑	☑	☑	☐	☐
Criteria:					
or:					

▲ Figure 19.15 Using the Totals row in the query grid

When the query is run, the calculation results are shown. Figure 19.16 shows the query results, including the use of the Avg calculation to work out the average price by make.



Make	CountOfMake	SumOfPrice	AvgOfPrice
BMW	1	£24,375.00	£24,375.00
Fiat	1	£15,250.00	£15,250.00
Ford	3	£40,599.00	£13,533.00
Hyundai	1	£18,250.00	£18,250.00
Kia	3	£49,000.00	£16,333.33
Toyota	1	£12,375.00	£12,375.00
VW	2	£30,490.00	£15,245.00

▲ Figure 19.16 The result of running the query using different calculations

As well as using Count, Sum and Average from the Query Totals drop down, you can also select Min and Max functions as well as some statistical calculations.

OUTPUTTING DATA

There are three basic methods of outputting data that you have already used.

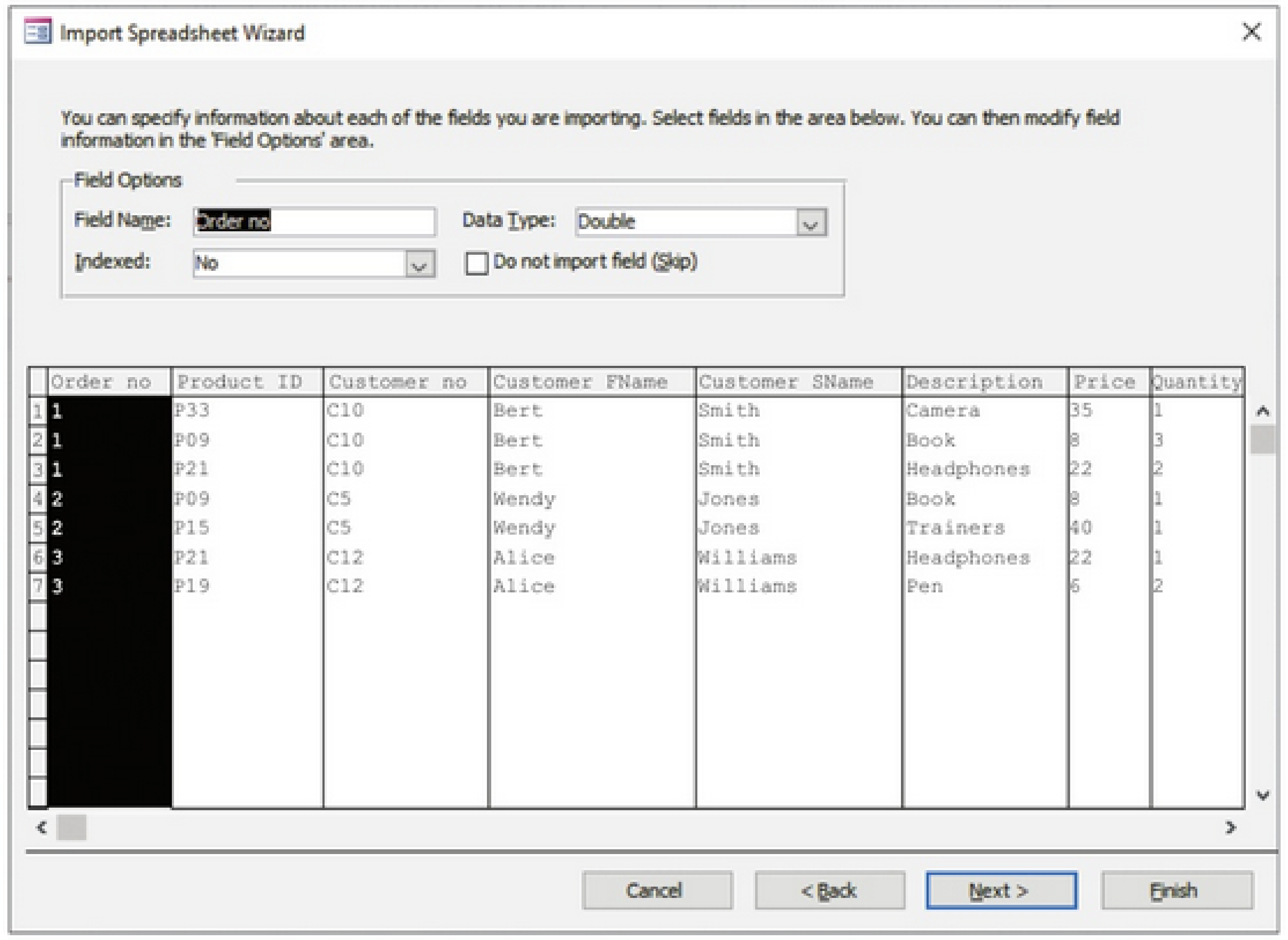
Datasheet view shows all the records in a table, or all the matching records from a query. Datasheet view is a simple and compact way of showing a lot of data and has built in features such as 'filter' and 'find'. However, it is not very user friendly or customisable.

Forms are really designed to show one record at a time, although you can create a form layout that shows multiple records. Designing and laying out forms can be awkward and time consuming. However, you can customise forms to provide features that make them easy for end users to carry out tasks, such as action buttons, onscreen help and so on. The data that is shown in a form can be based on a table or on a query. Forms are also ideal for inputting data because of the many validation and help features you can add.

Reports are customisable and can be based on tables or queries, but they can be awkward to design and create. Reports are designed for printing and displaying multiple records.

IMPORTING DATA

As already mentioned, a database might be developed to replace a previous system that is based on spreadsheets or other software. Importing data from spreadsheets is fairly straightforward as Access provides a Wizard to guide you through the process. To import data, first go to the External Data menu and on the left side of the toolbar choose New Data Source; this will drop down a menu from where you can choose From File and then the type of file you want to import, an Excel file in this example. You then need to select the spreadsheet you want to import and choose if you want to save the data in a new table or append (add) them to an existing table. The Import Wizard will then display a preview of the spreadsheet data and you can choose if the first row of the spreadsheet contains the column headings (field names) for the data. Next it will present each column or field so you can check the data type that the Wizard has chosen. Figure 19.17 shows the Import Spreadsheet Wizard.



You can specify information about each of the fields you are importing. Select fields in the area below. You can then modify field information in the 'Field Options' area.

Field Options
Field Name: Order no
Data Type: Double
Indexed: No
☐ Do not import field (Skip)

	Order no	Product ID	Customer no	Customer FName	Customer SName	Description	Price	Quantity
1	1	P33	C10	Bert	Smith	Camera	35	1
2	1	P09	C10	Bert	Smith	Book	8	3
3	1	P21	C10	Bert	Smith	Headphones	22	2
4	2	P09	C5	Wendy	Jones	Book	8	1
5	2	P15	C5	Wendy	Jones	Trainers	40	1
6	3	P21	C12	Alice	Williams	Headphones	22	1
7	3	P19	C12	Alice	Williams	Pen	6	2

▲ Figure 19.17 Import Spreadsheet Wizard

The next page of the wizard asks you to accept or choose a primary key for the table, and finally asks you to set the name of the table to import the data into. The data will then be imported and a table created (if that is what you selected).

If the data to be imported is in a text file such as a **comma separated value (CSV)** or TXT (plain text), you will need to tell the Wizard which delimiter (such as a space or comma) is used to separate the fields and records in the file. If the file you are importing is in CSV format then the delimiters should be commas.

KEY TERM

comma separated value (CSV): a text data file where the data values are separated by commas

EXPORTING DATA

You can also export data from a table into a variety of formats. Select the External data menu and then, in the Export section in the centre of the toolbar, click the Excel icon to export a table to an Excel file, or the Text icon to export it as text. You can also export a table to a PDF document, which will create a PDF containing a table with the data displayed. Exporting a table to HTML (web page) format is also supported. Click the More icon on the right side of the External Data toolbar and select HTML document from the list that drops down. This will create an HTML document with the data in a table format that can be displayed in a browser.

EVALUATING DATABASE TOOLS

Choosing the best tool for a particular requirement will depend on the skill level of the user and what task needs to be completed. Tools such as queries provide many sophisticated features, but they are complex to use and not really suited to end users for this reason. Forms (primarily for data input) and Reports for data output, which might be based on queries and have been built by an IT professional to meet the end-user needs, are easier to use and can provide the basis for creating an end-user application.

The suitability of any database tool really needs to be evaluated in the context of the scenario, and the level of the users' skills with databases. The key questions that need to be considered are:

Does the database feature do what the user needs it to do? This might be input data, find data that matches certain criteria or output data in an understandable way.

Will the user find the database features easy to use, bearing in mind their skill level?

You might also consider how the feature could be improved and made easier to use. This might include things like providing the user with a menu that has buttons to run various queries, or using a chart on a report rather than a table of figures.

SKILLS

PROBLEM SOLVING

ACTIVITY 3

1. Use the library database from Activity 2 to create a report, based on a query that lists all borrowers in alphabetical order, showing the title and author of the books they have on loan.
2. Practise exporting and importing data by exporting one of the data tables you have created in Access to a spreadsheet and also to a CSV text file. Then import both the spreadsheet and text file into another Access database.

KEY TERM

dashboard: an interface that allows a user to choose various ways of presenting relevant information

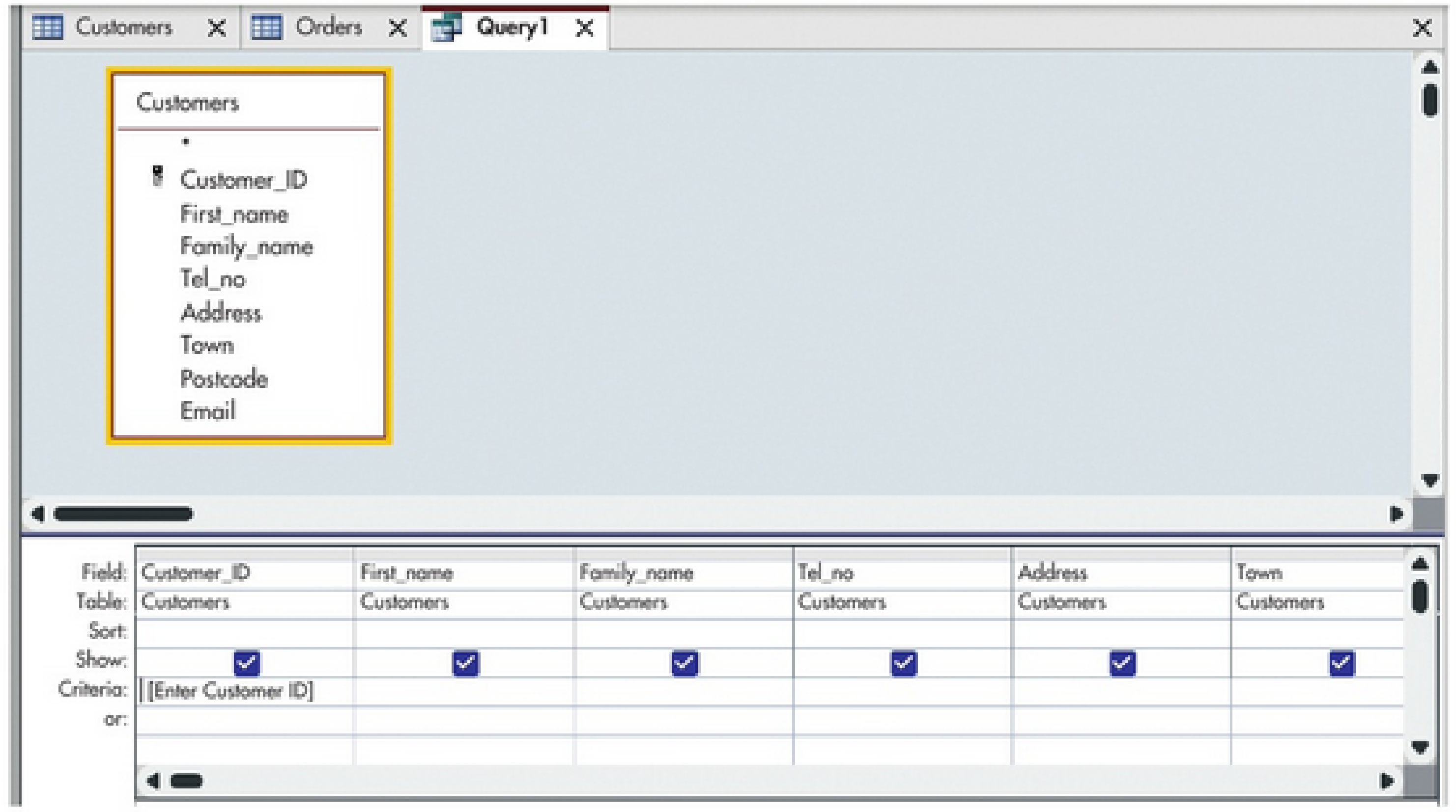
USER INTERFACE

To create an end-user database application you need to provide a complete easy-to-use form-based user interface that includes features such as menus, **dashboards** and action buttons. You have looked at some of these features already, and you will now look at them in further detail to see how to combine them into a complete application.

DATA ENTRY FORMS

In Topic 18, you saw how a data entry form can be created to add new records; but how can users select and amend existing records? The user could go through all the records in a table using the next/previous record controls that appear at the bottom of a form, but it would be much better to provide a method for them to view and amend a specific record. Imagine a customer service application in a call centre where a customer might call up to change their address. The call centre personnel will want to be able to go straight to a customer's record, probably by entering the customer's ID or other details. One way of achieving this is to use a query with a parameter that causes an input box to be presented to the user, and then uses their input to filter the results. Creating a form that uses this query as its data source will mean the query results will be displayed in a form.

Let's demonstrate how this could work. The Orders database you have used before has a table for customers, and as mentioned above, a complete Database application would probably need a feature for the users to look up an individual customer's record and amend it. The first step in creating this feature would be to create a query that uses a parameter as a criterion to filter the table on its key field, the customer ID. Figure 19.18 shows the query grid with the parameter set.



Field:	Customer_ID	First_name	Family_name	Tel_no	Address	Town
Table:	Customers	Customers	Customers	Customers	Customers	Customers
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:	[Enter Customer ID]					
or:						

▲ Figure 19.18 A query using a parameter as a criterion

When this query runs, an input box will pop up and display the message inside the brackets (Enter Customer ID). The customer record with a matching ID will then be displayed. You should save the query with a meaningful name, such as Find_Customer.

The next step is to create a form that uses this query as its data source. The Form Wizard can be used, adding all the fields from the query with the default settings. When the form opens, the input box from the query will display and once a valid Customer ID is entered, the customer details will appear in the form.

You should customise the form in several ways.

Disable the Customer ID field so the user cannot change it. Since the field is used as a link to the Orders table it must not be changed or any existing orders from that customer will be 'lost' (unless you have turned on the 'Cascade update related fields' feature as described earlier). Changing a customer ID would also cause other problems – for example, the customer details printed on the form sent out with the order would be wrong.

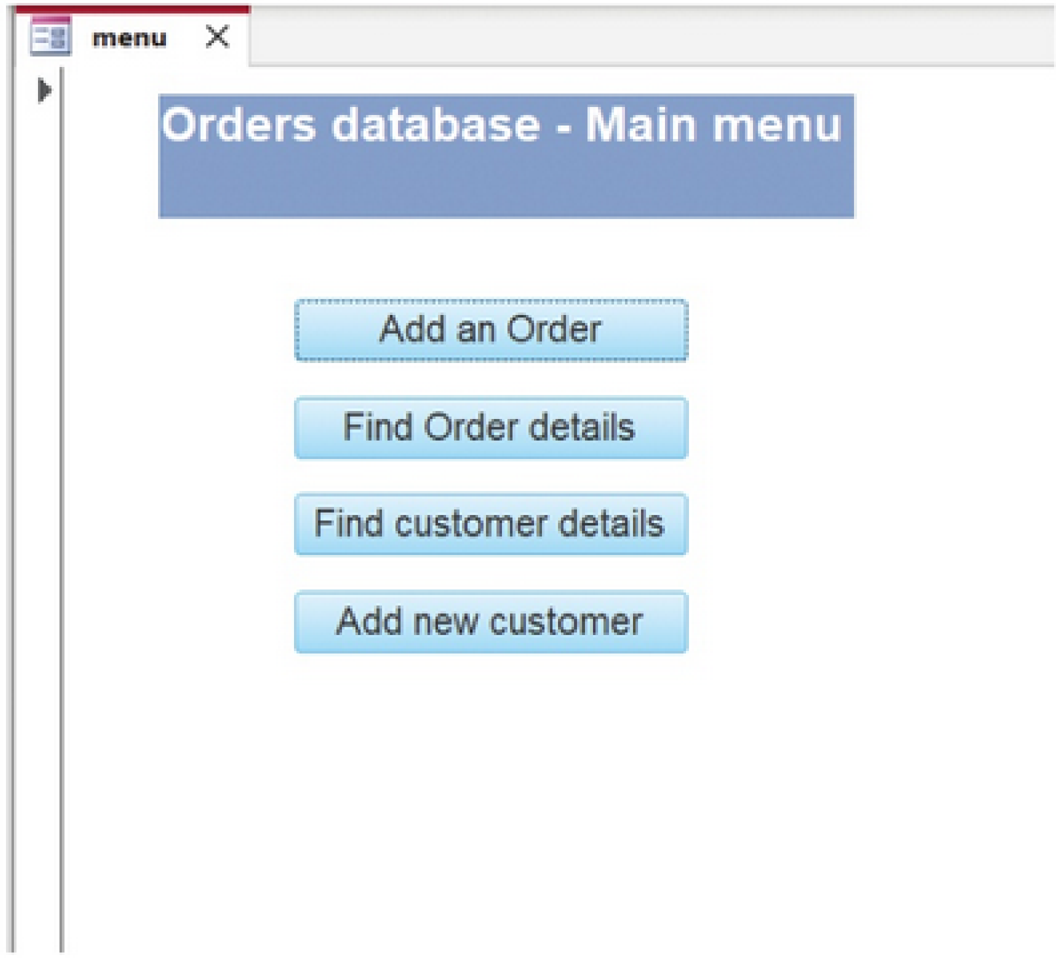
Add a Save button with a confirmation message as well as a Cancel button. You saw how to do this with command buttons and macros in Topic 18.

Add validation where possible, which you also covered in Topic 18. Data such as names and addresses are difficult to validate so there may not be much you can do. Real-life applications often do postcode validation using a postcode data file provided by the postal service, but this may not be available in some countries. You should always try to validate email addresses, as they must always contain the @ symbol as a minimum.

MENUS AND DASHBOARDS

To build an end-user database application with Access you will need a main menu where the user can select forms to add and amend records. You can do this by creating a form that is not linked to a table and then adding command buttons that use the 'Form open' action to open the required forms when the user clicks them.

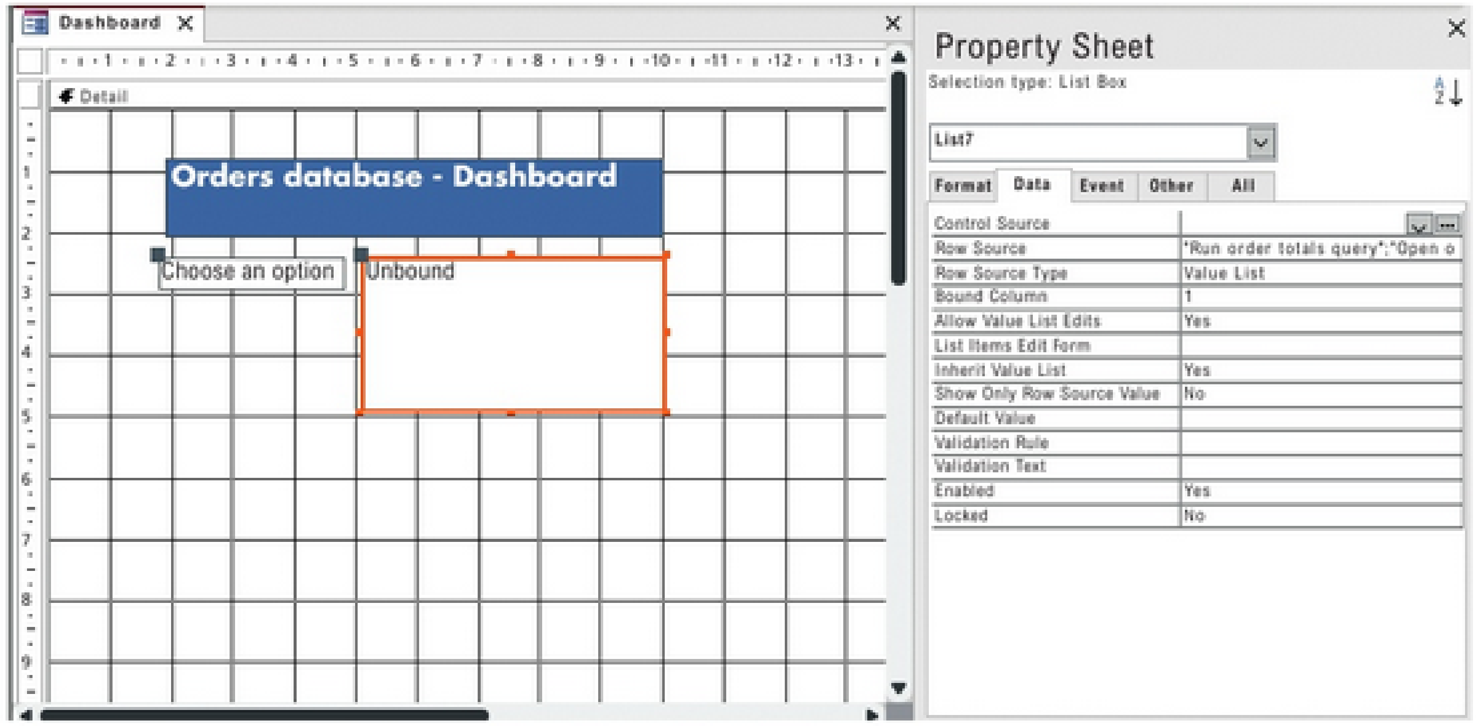
To create a form that is not linked to a table use the Form Design icon in the Create menu. This will open a blank form in Design view. Add titles to the form using the Label control in the Form Design toolbar and then use the Command Button control. Using the Button Wizard, you can select the Form Operations category and the Open Form action. Figure 19.19 shows an example of a menu created this way.


menu
Orders database - Main menu
Add an Order
Find Order details
Find customer details
Add new customer

▲ Figure 19.19 A main menu form

A dashboard is a type of menu, although a dashboard is typically aimed at providing relevant information to users rather than allowing them to input data.

Using buttons isn't the only way you can allow users to select options on a dashboard or menu. Another way of providing the functionality to select different options is to use a list box. You can create a list box on a blank form using the List Box Wizard and enter the options you want to be shown in the List box. To make the list box open queries, forms or reports you need to create a macro that will run when the user clicks an option in the list box. Figure 19.20 shows a dashboard form in Design view with the list box created.


Dashboard
Detail
Orders database - Dashboard
Choose an option
Unbound
Property Sheet
Format
Data
Event
Other
All

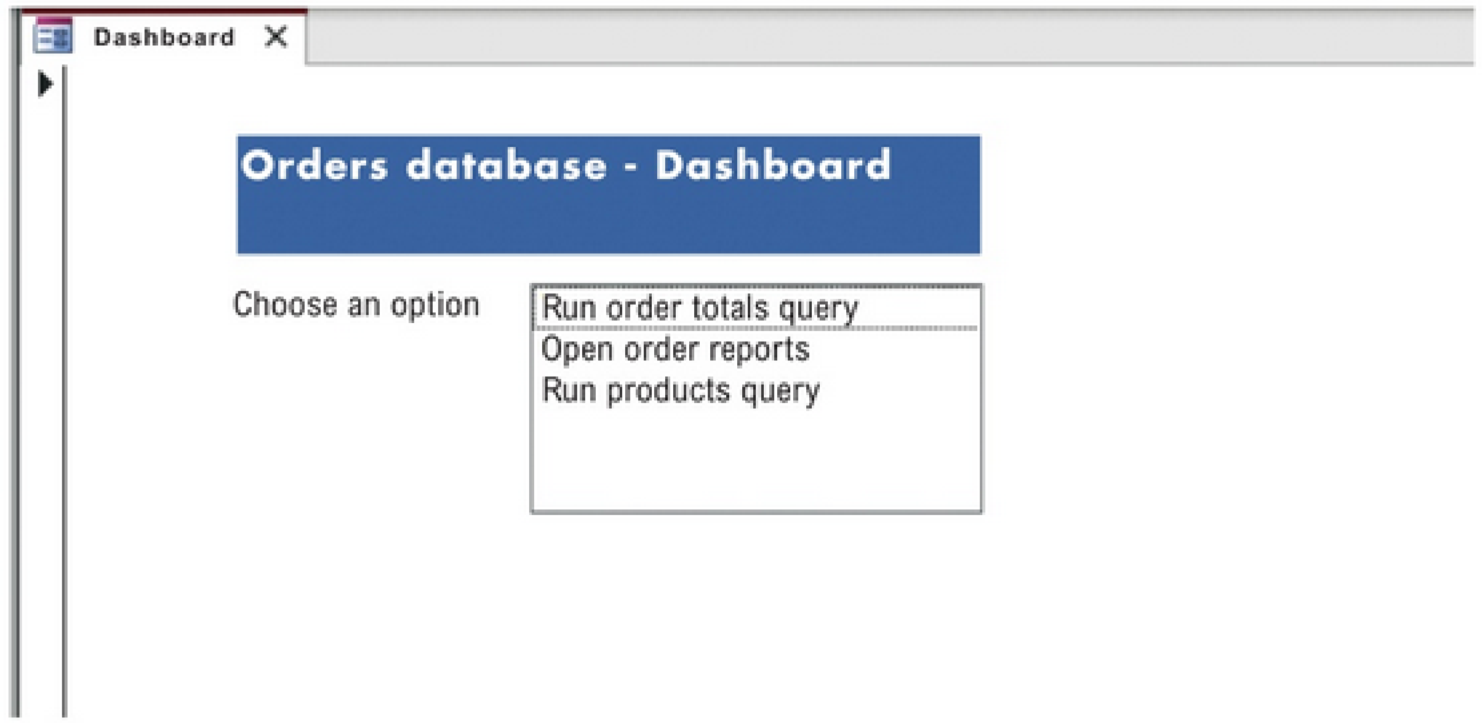
Selection type: List Box

List7

Control Source	
Row Source	"Run order totals query";"Open o
Row Source Type	Value List
Bound Column	1
Allow Value List Edits	Yes
List Items Edit Form	
Inherit Value List	Yes
Show Only Row Source Value	No
Default Value	
Validation Rule	
Validation Text	
Enabled	Yes
Locked	No

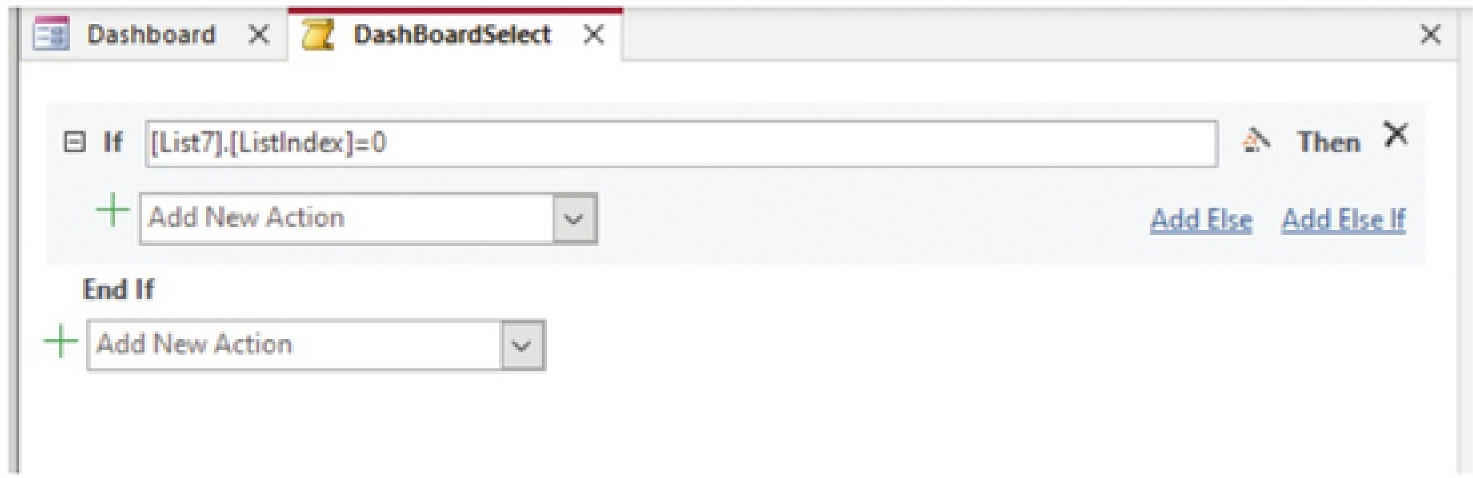
▲ Figure 19.20 A dashboard list box in Design view

The macro that you write will need to find the option that the user clicked in the list box and then open the corresponding query or report. The list box is shown in Figure 19.21.


Dashboard
Orders database - Dashboard
Choose an option
Run order totals query
Open order reports
Run products query

▲ Figure 19.21 The dashboard form in Form view

The macro will use an 'if' statement to test the value in the List Index property. Go to the Create menu and choose Macro. Click the Add New Action drop down and choose if. Then in the box next to the 'if' command, enter the instruction shown in Figure 19.22.

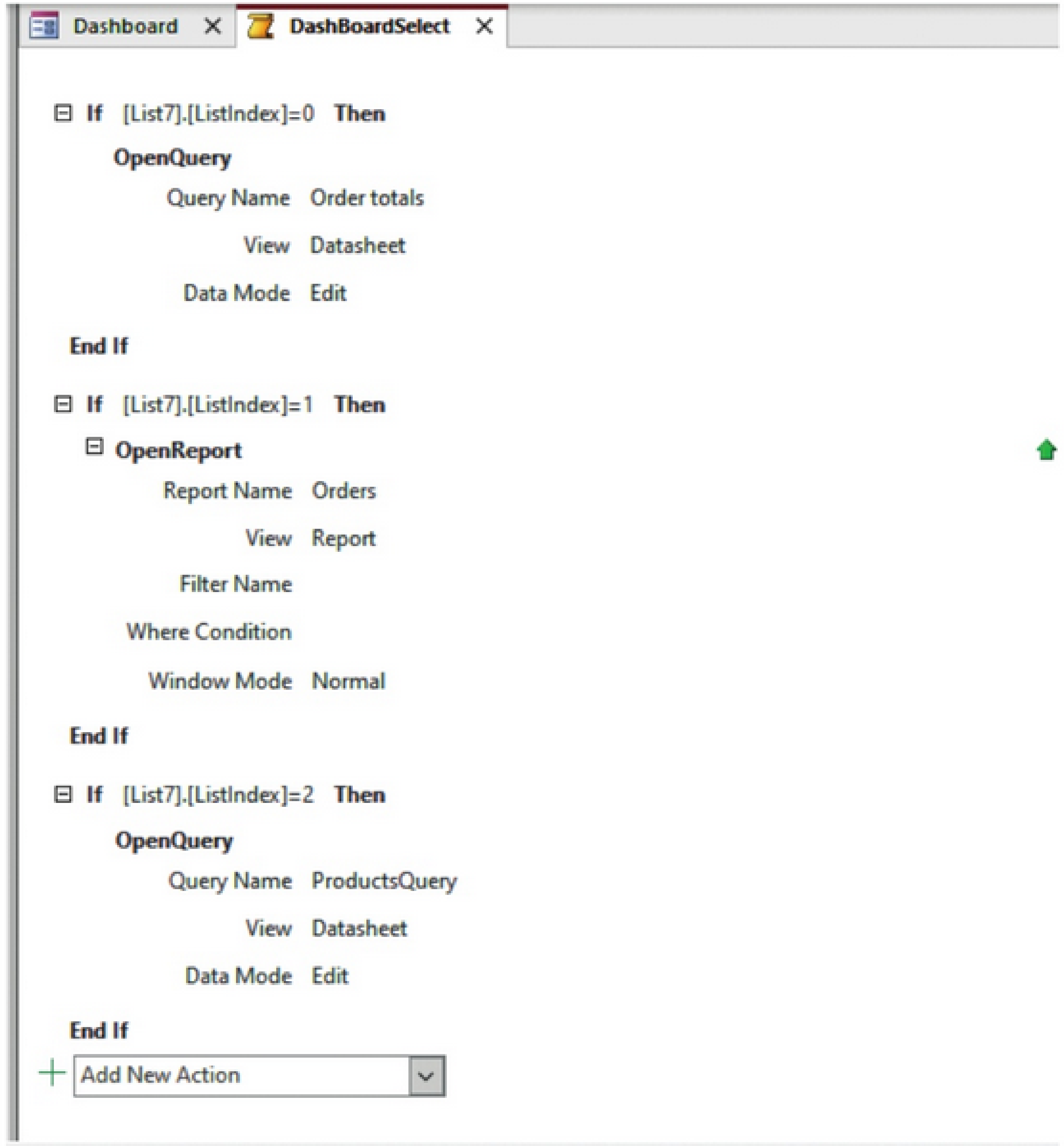

Dashboard
DashBoardSelect
Add New Action
Add Else
Add Else If
Add New Action

```
If [List7].[ListIndex]=0 Then
    Add New Action
End If
```

▲ Figure 19.22 Entering the 'if' command

In the Add New Action drop down directly under the 'if' command choose the command `Open Query` and then set the query name to the Order totals Query.

You need to repeat the 'if' command for each option in the list box. The complete macro is shown in Figure 19.23.

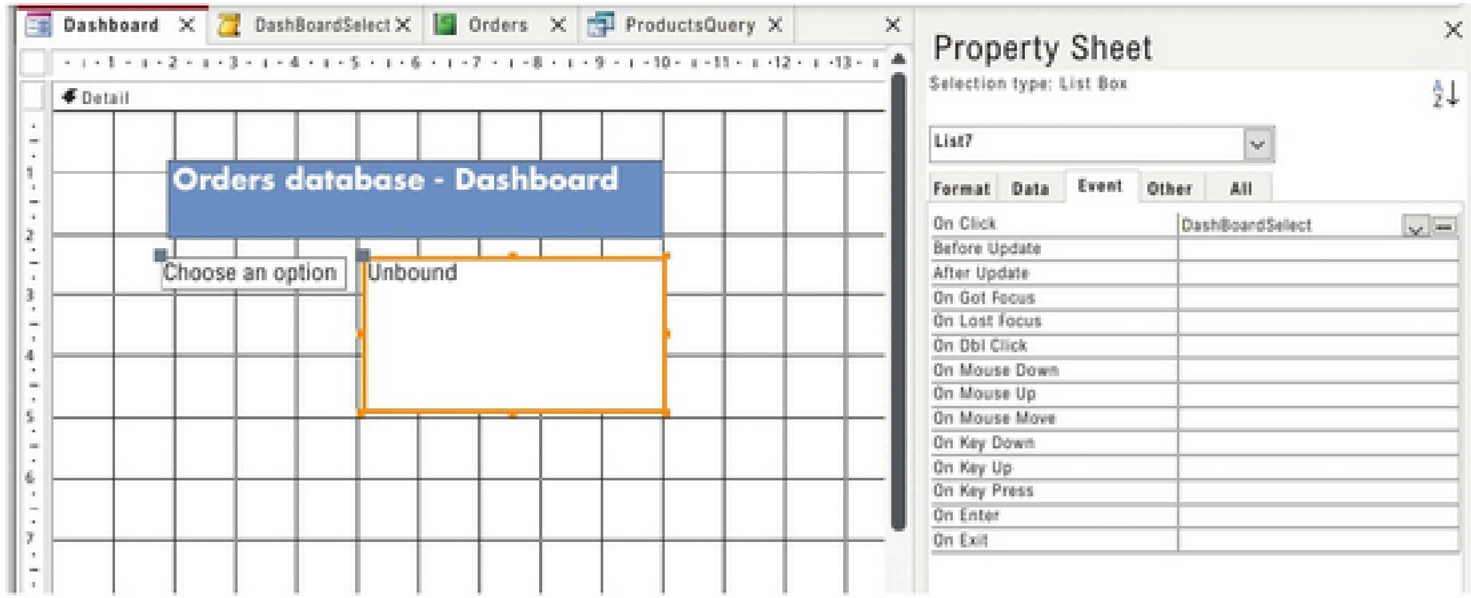

Dashboard
DashBoardSelect

```
If [List7].[ListIndex]=0 Then
    OpenQuery
        Query Name  Order totals
              View  Datasheet
         Data Mode  Edit
End If
If [List7].[ListIndex]=1 Then
    OpenReport
         Report Name  Orders
                View  Report
         Filter Name
     Where Condition
         Window Mode  Normal
End If
If [List7].[ListIndex]=2 Then
    OpenQuery
        Query Name  ProductsQuery
              View  Datasheet
         Data Mode  Edit
End If
Add New Action
```

▲ Figure 19.23 The completed macro to make the dashboard options open the required query or report

The macro should be saved and given a meaningful name; in this example it is named DashBoardSelect.

The final step is to link the query to the 'On Click' event for the list box. This event needs to trigger the macro when the user clicks one of the list box options. Figure 19.24 shows the macro linked to the On Click event in the Property sheet for the list box.


Dashboard
DashBoardSelect
Orders
ProductsQuery
Detail
Orders database - Dashboard
Choose an option
Unbound
Property Sheet
Selection type: List Box
List7
Format
Data
Event
Other
All
Before Update
After Update
On Got Focus
On Lost Focus
On Dbl Click
On Mouse Down
On Mouse Up
On Mouse Move
On Key Down
On Key Up
On Key Press
On Enter
On Exit

On Click: DashBoardSelect

▲ Figure 19.24 The Macro link to the On Click event for the list box

ACTION BUTTONS

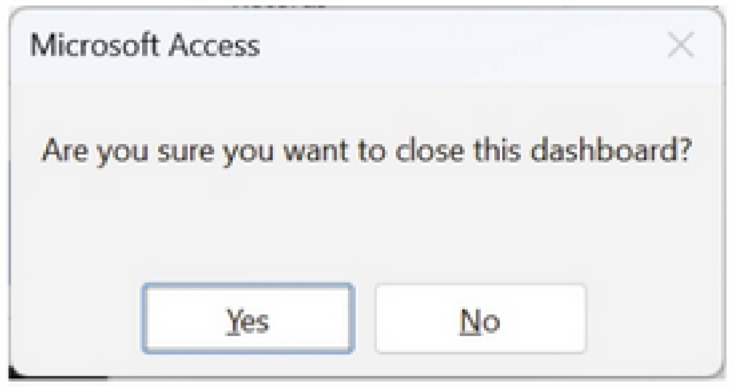
You have already seen some examples of the use of action buttons. The Button Wizard in the Form Design view allows you to easily create a range of different buttons that can do things like close or open a form, go to the first, last or next record, and so on. Where you want to create a custom action you can write a macro and attach the macro to a button using the 'Miscellaneous' category in the Button Wizard (or the On Click property of a list box as shown above).

An example of a macro that may be needed in an application is a confirm message to ask a user if they are sure they want to carry out some action (although you should not overdo the use of confirm requests as users can find them frustrating). Let's see how you can provide a button to close the dashboard created above and then ask the user if they are sure they want to close the dashboard.

This uses the MsgBox function, which allows you to display various buttons in a message box and returns a value depending on which button the user clicks. You select the buttons to appear in the message box using a parameter. For example, this MsgBox function:

MsgBox("Are you sure you want to close this dashboard?",4)

uses the button type value of 4. This displays 'Yes' and 'No' buttons with the message in quotes shown in the box. This will produce a message box that looks like Figure 19.25.


Microsoft Access
Yes
No

Are you sure you want to close this dashboard?

▲ Figure 19.25 A message box with Yes and No buttons

Other values you can use in a MsgBox function include:

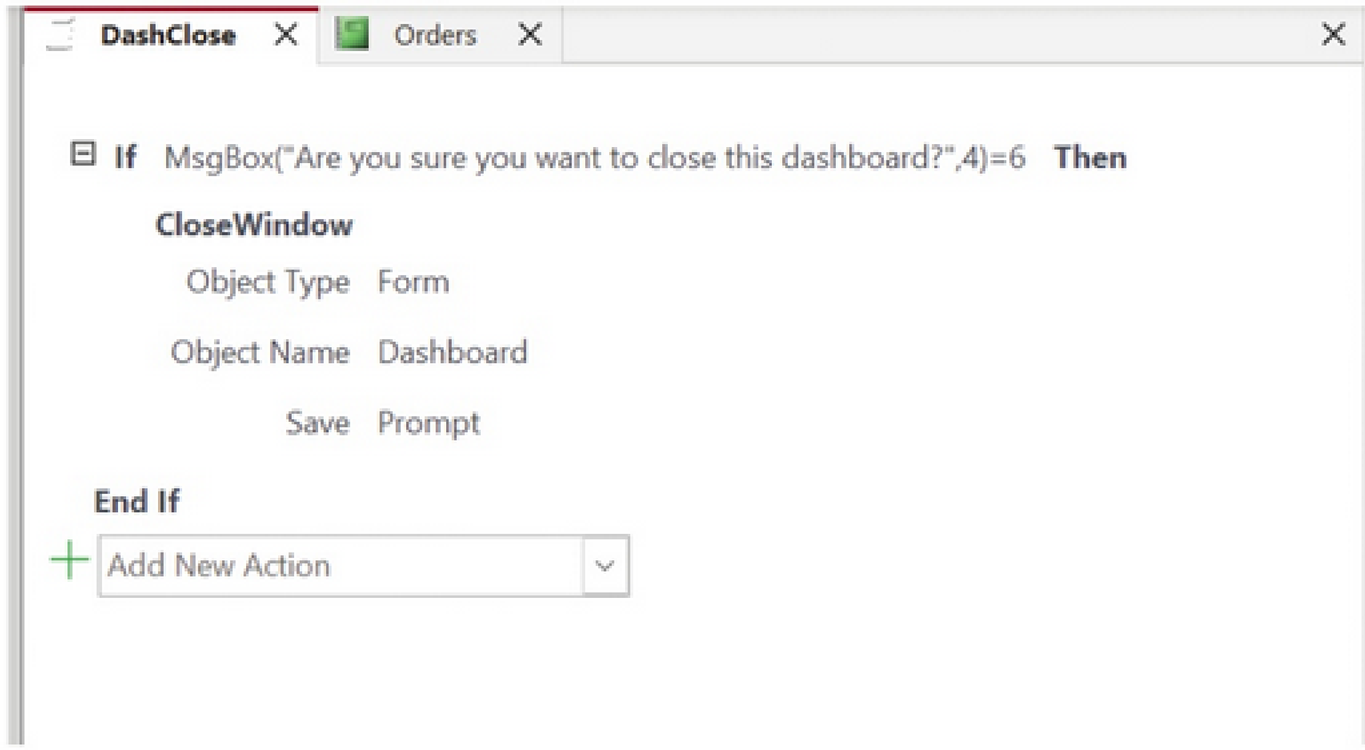
0 – an OK button only

1 – OK and Cancel buttons

2 – Abort, Retry, and Ignore buttons

3 – Yes, No, and Cancel buttons

You also need to find out which button the user has clicked. If they click the No button, you don't need to do anything; if they click the Yes button, the dashboard form should close. The macro to do this is shown in Figure 19.26.



```
If   MsgBox("Are you sure you want to close this dashboard?",4)=6   Then
     CloseWindow
        Object Type   Form
        Object Name   Dashboard
               Save   Prompt
End If
```

▲ Figure 19.26 A macro using the MsgBox function to display a confirm message

As mentioned earlier, the MsgBox function returns a value that indicates which button has been clicked. If the Yes button is clicked it returns the value of 6. The other values that can be returned (although some don't apply to the yes/no button) include:

1. OK button clicked
2. Cancel button clicked
3. Abort button clicked
4. Retry button clicked
5. Ignore button clicked
6. Yes button clicked
7. No button clicked

You can use the MsgBox function to provide a number of different ways to interact with the user and carry out actions.

SYSTEM OUTPUTS

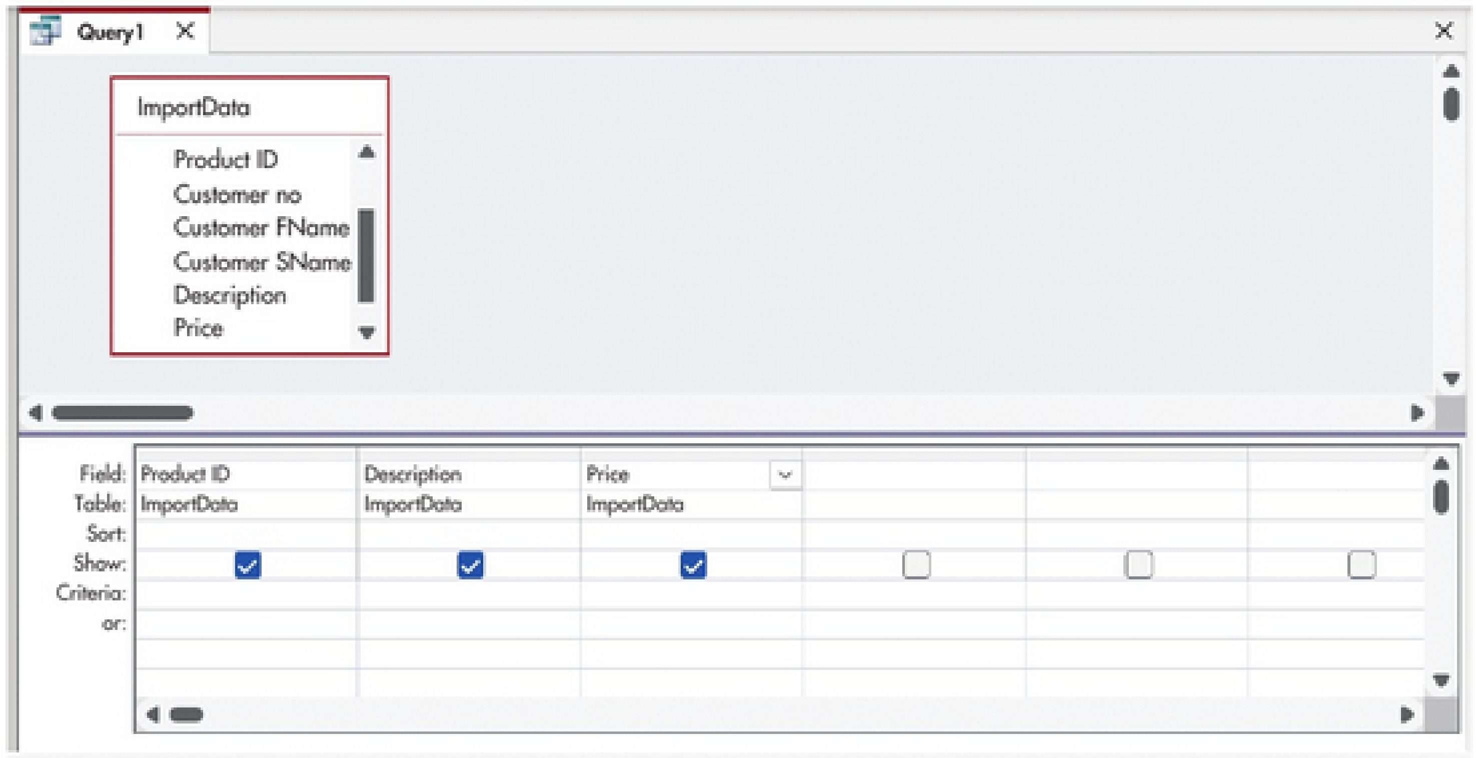
You have already seen how you can output data from a query in the Datasheet view, and how reports can format and output data from table and queries. Here you are going to look at some other output techniques.

Data tables

In some applications you may need to use a query to make a new table or to amend data in an existing table. For example, you may need to import data as a text file or spreadsheet (as described earlier) and then modify that data to fit in with the database. Text and spreadsheet files are of course 'flat' files with no relationships, so you may need to split data into several tables that can be related.

Earlier you saw how you could import data from a spreadsheet, and you used an example of some order data. Suppose now that you wanted to extract data from there to create a Products table.

You can use an Action query to split it into the required tables. It's a good idea to start off with a standard select query and make sure it is selecting the correct records before you convert it into a Make Table query. To create the Products table, first go to the Create menu and click the Query Design icon, then add the table created from the imported data. For the Products table, the only fields we need are Product ID, Description and Price. Drag these into the query grid as shown in Figure 19.27.

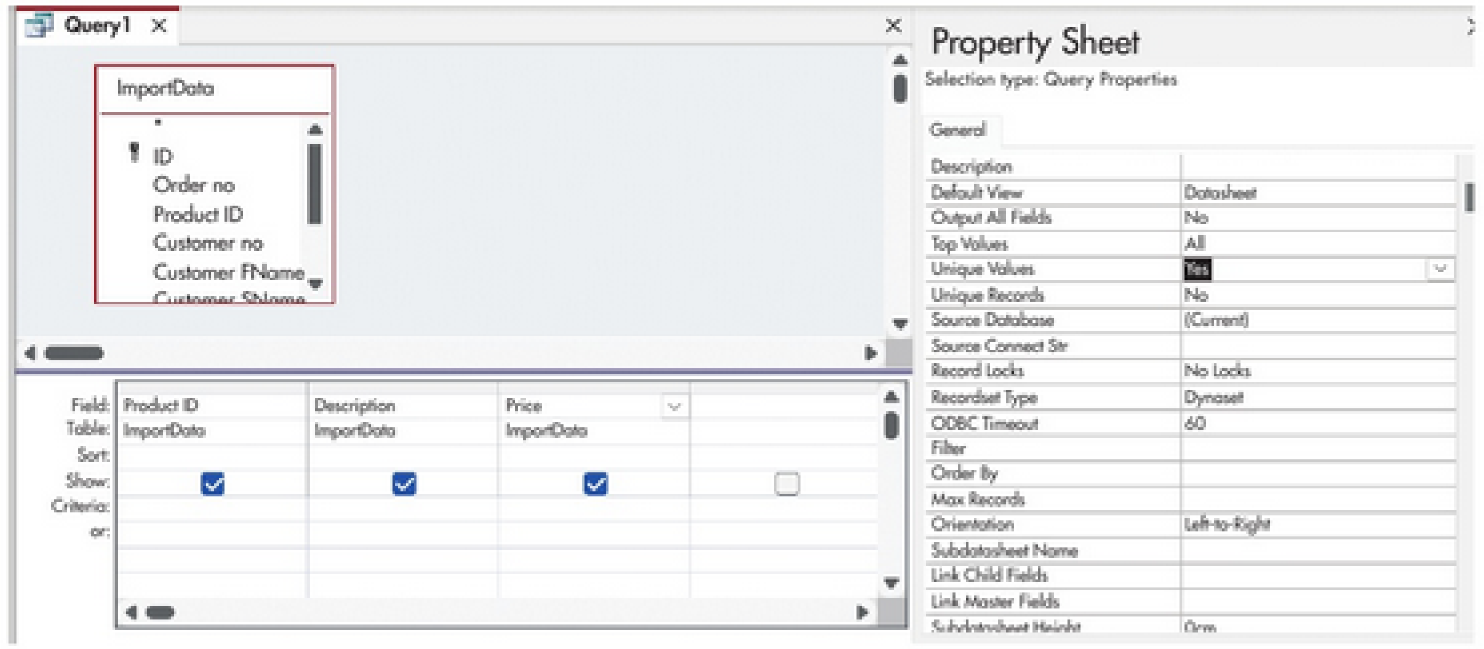


ImportData
Product ID
Customer no
Customer FName
Customer SName
Description
Price

Field:	Product ID	Description	Price			
Table:	ImportData	ImportData	ImportData			
Sort:						
Show:	☑	☑	☑	☐	☐	☐
Criteria:						
or:						

▲ Figure 19.27 The query grid to create a new Products table

If you run this query you will see that some products appear multiple times (because they have been ordered on different orders). To create the Product table, you need each product to occur only once in the query results. To show only unique values you need to click in the grey background in the top part of the query screen, which will display the query properties in the right side panel. The fourth property down is called Unique Values. Change the default value of No to Yes, as shown in Figure 19.28.


Query1
ImportData
*
ID
Order no
Product ID
Customer no
Customer FName
Customer SName
Property Sheet
Selection type: Query Properties
General

Field:	Product ID	Description	Price	
Table:	ImportData	ImportData	ImportData	
Sort:				
Show:	☑	☑	☑	☐
Criteria:				
or:				

Description	
Default View	Datasheet
Output All Fields	No
Top Values	All
Unique Values	Yes
Unique Records	No
Source Database	(Current)
Source Connect Str	
Record Locks	No Locks
Recordset Type	Dynaset
ODBC Timeout	60
Filter	
Order By	
Max Records	
Orientation	Left-to-Right
Subdatasheet Name	
Link Child Fields	
Link Master Fields	
Subdatasheet Height	0cm

▲ Figure 19.28 Changing the 'Unique Values' property

Now when the query is run, each product is only displayed once. You can now convert this query into a Make table query and use it to create the Products table. To do this, click the Make table icon in the toolbar. Access will then ask you the name of the table to create, which in this case is Products. The query will then run and a warning message will appear to tell you that the records will be added to a table. Clicking yes will insert the records in the newly created table. You should only run a Make table query once, as each time it is run it will add the records to the table again.

Reports

As well as the basic reports you have already looked at, you can create more sophisticated reports using charts to show data in a graphical way. To add a chart to a report you first need to create a report and then change to Design view. The report needs to be based on a table or query that you can draw a chart from. There are several different types of chart you can create.

Pie chart

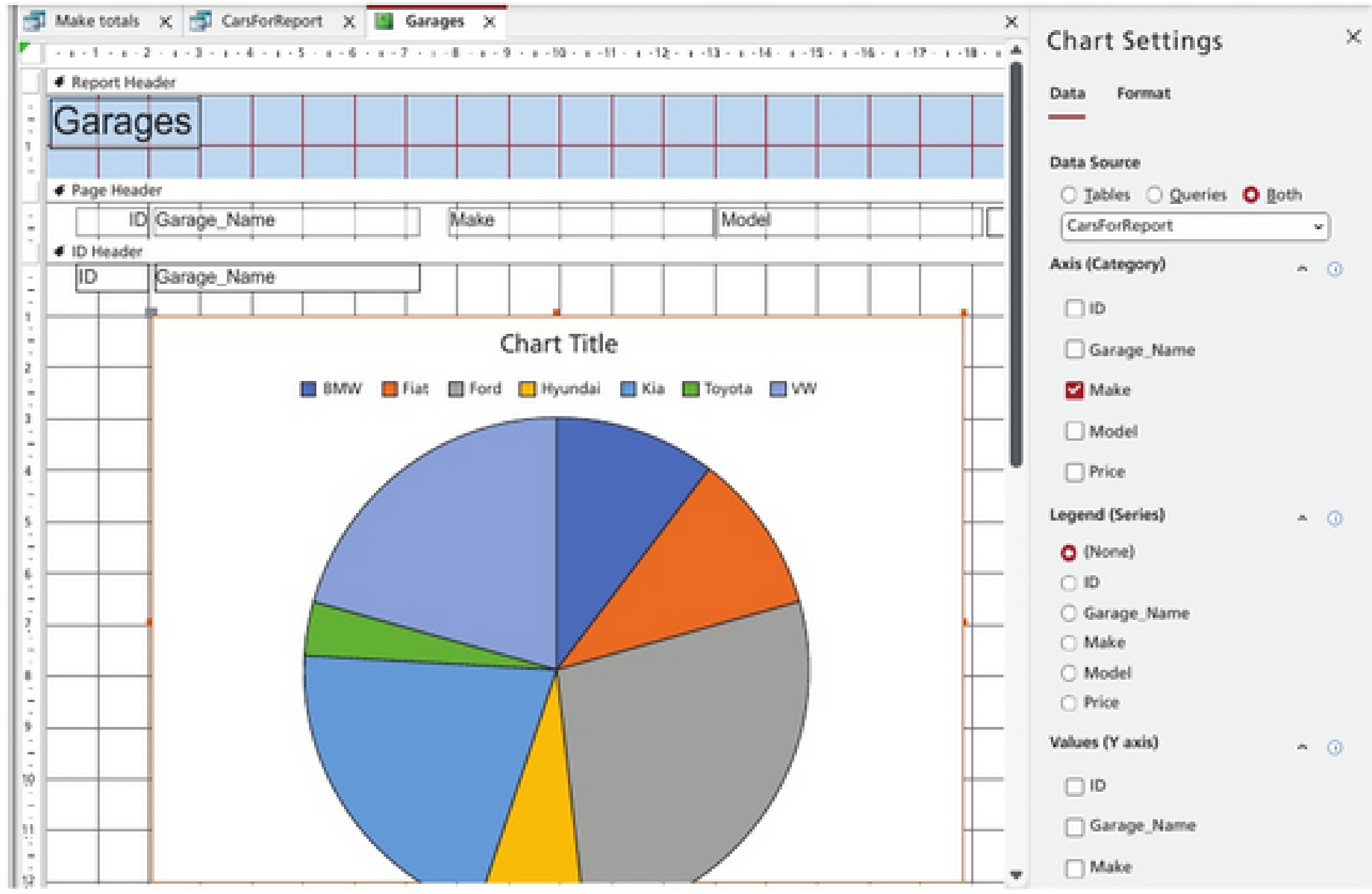
This is a circular chart that has segments (or slices) to represent a proportion of a total amount. You only need one data set to make a pie chart. For example, using the Cars for Sale database you could create a query that adds up the number of cars in stock of each make. We did this earlier so we can use that query and create a simple report. To add a pie chart, open the report in Design view and make the Detail section in the centre of the report much bigger so there is room for a chart. Then, from the Controls section in the Toolbar, choose the Insert Modern Chart icon, then drag out a large box in the centre of the report. This will open the Chart setting in a pane on the right side of the design window. This is shown in Figure 19.29.

You can change the title of the pie chart by opening the property sheet (click the Property sheet icon in the toolbar) and then, in the Format properties, changing the Chart Title property.

Column chart

This has vertical bars that represent data values. You can create a bar chart by choosing it from the drop down that appears when you click the Insert Modern Chart icon. You can also change an existing chart to a column chart by changing the Chart Type property in the Format section of the Property sheet. With a Column chart you can select the data category, which is displayed along the 'x' (bottom) axis of the chart, the series (which defines the bars in the chart) and the 'y' axis, which runs up the left side of the chart. Figure 19.30 shows an example.

As well as pie and column charts, you can also select bar charts (with horizontal bars) and line charts.


Make totals
CarsForReport
Garages
Report Header
Garages
Page Header
ID
Garage_Name
Make
Model
ID Header
ID
Garage_Name
Chart Title
BMW
Fiat
Ford
Hyundai
Kia
Toyota
VW
Chart Settings
Data
Format

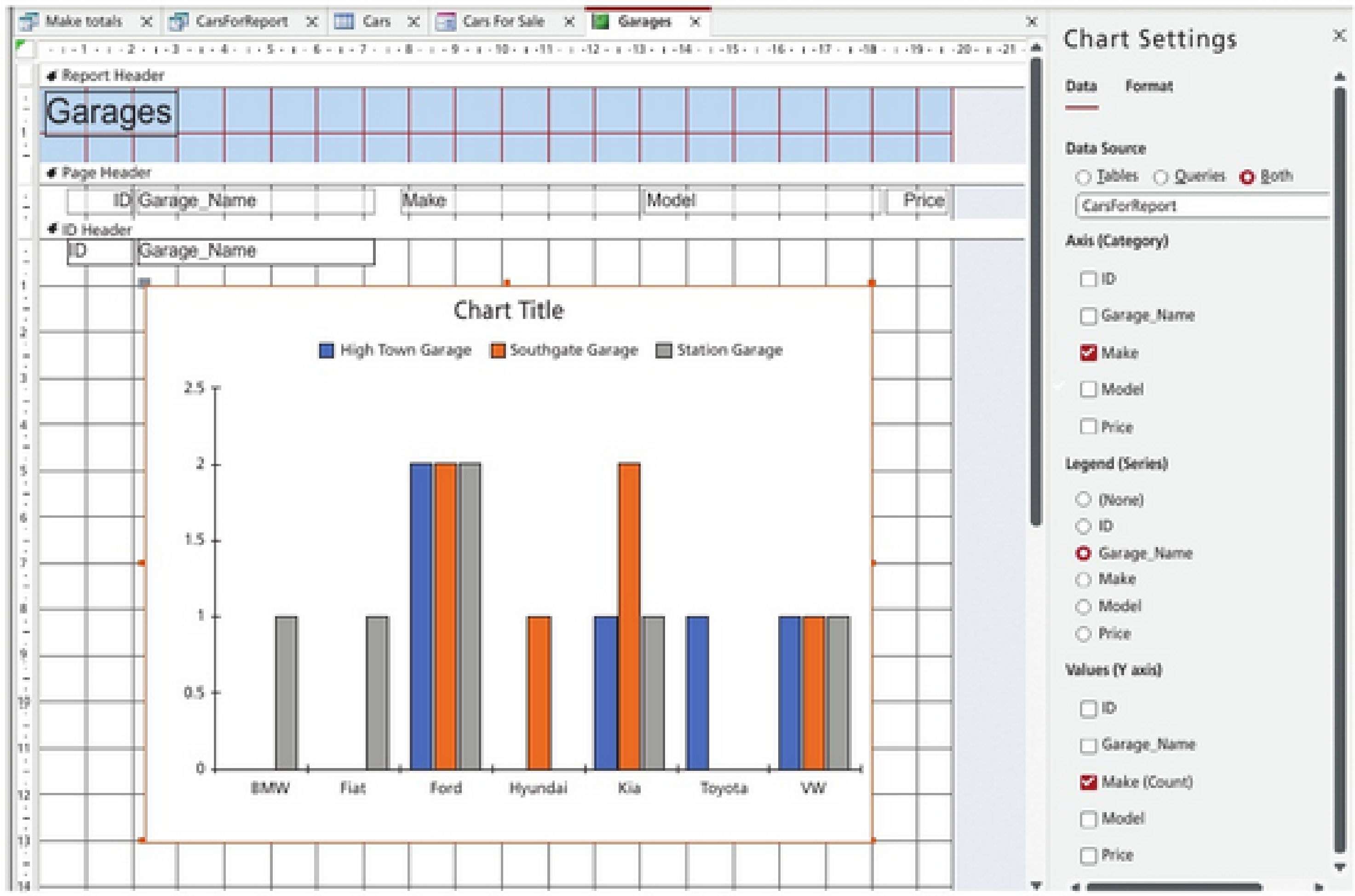
Data Source
○ Tables ○ Queries ● Both
CarsForReport

Axis (Category)
☐ ID
☐ Garage_Name
☑ Make
☐ Model
☐ Price

Legend (Series)
● (None)
○ ID
○ Garage_Name
○ Make
○ Model
○ Price

Values (Y axis)
☐ ID
☐ Garage_Name
☐ Make

▲ Figure 19.29 Adding a pie chart to a report


Make totals
CarsForReport
Cars
Cars For Sale
Garages
Report Header
Garages
Page Header
ID
Garage_Name
Make
Model
Price
ID Header
ID
Garage_Name
Chart Title
High Town Garage
Southgate Garage
Station Garage
2.5
2
1.5
1
0.5
0
BMW
Fiat
Ford
Hyundai
Kia
Toyota
VW
Chart Settings
Data
Format

Data Source
○ Tables ○ Queries ● Both
CarsForReport

Axis (Category)
☐ ID
☐ Garage_Name
☑ Make
☐ Model
☐ Price

Legend (Series)
○ (None)
○ ID
● Garage_Name
○ Make
○ Model
○ Price

Values (Y axis)
☐ ID
☐ Garage_Name
☑ Make (Count)
☐ Model
☐ Price

▲ Figure 19.30 A column chart

Error messages

It is again important that you add error messages to help the user understand any issues that arise. You should therefore use the Validation Text property to provide a meaningful error message when adding validation rules to fields.

SKILLS

PROBLEM SOLVING

ACTIVITY 4

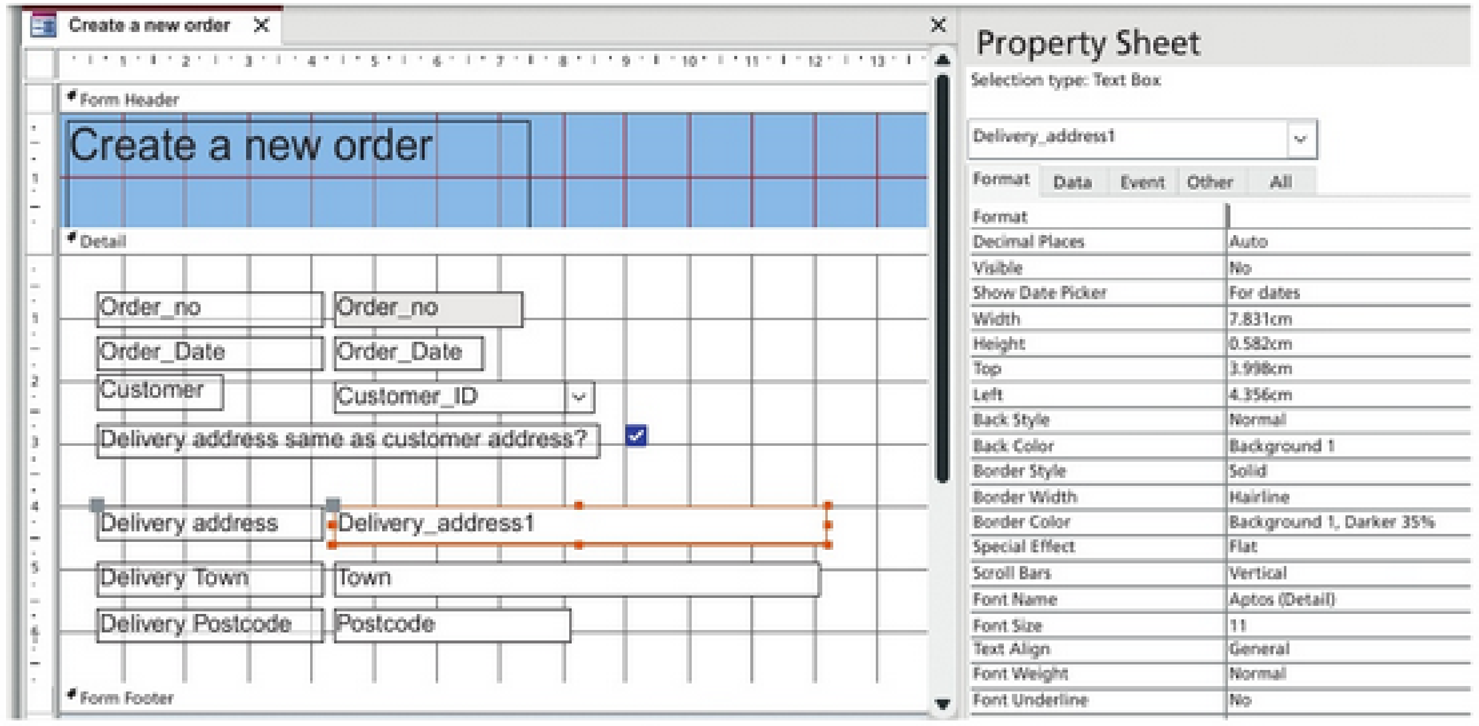
1. Use the library database to create a parameter query that will ask for the Borrower ID and then list the books they have on loan.
2. Make a menu or dashboard for the library database. This should allow a user to open the query you created above, and the ones you created in Activity 3.
3. Create a report for the library database and add a pie chart that shows how many books there are in each category.

USER INTERFACE COMPONENTS

You have already looked at a variety of user interface components that you can include in forms. Now you will look in a little more detail at some aspects of form design that can help to make an easy-to-use and professional interface.

Selecting/omitting fields. It is important that a user is presented with a form that contains the fields they need to complete, and does not show fields that they do not need to complete. In some applications it may be that the relevance of some fields will depend on previous choices that the user has made. Those optional fields should ideally be hidden until the user chooses something that will make them relevant.

For example, when placing an order, a company may have a different delivery address (for a branch office) to the main address (headquarters) of the company. If using a form, the delivery address could be hidden and a Boolean field used to indicate if the delivery address is the same as the main address. This would be represented on the form with a check box. If the user unchecks the box then that indicates that the delivery address and the main address are different, and the fields where the delivery address are entered should be made visible. Figure 19.31 shows the form in Design view.


Create a new order
Form Header
Create a new order
Detail
Order_no
Order_no
Order_Date
Order_Date
Customer
Customer_ID
Delivery address same as customer address?
Delivery address
Delivery_address1
Delivery Town
Town
Delivery Postcode
Postcode
Form Footer
Property Sheet
Format
Data
Event
Other
All

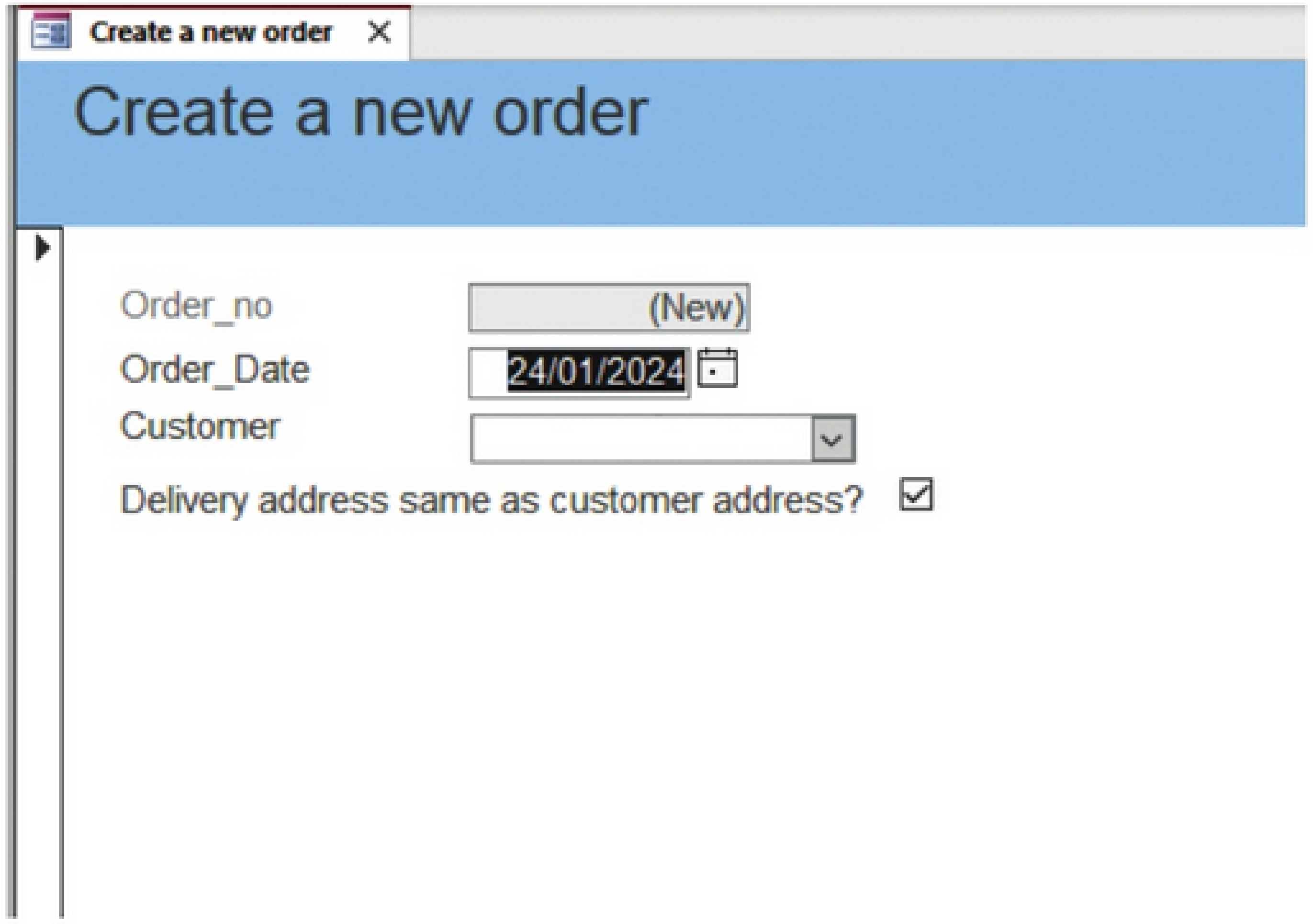
Selection type: Text Box

Delivery_address1

Format	
Decimal Places	Auto
Visible	No
Show Date Picker	For dates
Width	7.831cm
Height	0.582cm
Top	3.998cm
Left	4.356cm
Back Style	Normal
Back Color	Background 1
Border Style	Solid
Border Width	Hairline
Border Color	Background 1, Darker 35%
Special Effect	Flat
Scroll Bars	Vertical
Font Name	Aptos (Detail)
Font Size	11
Text Align	General
Font Weight	Normal
Font Underline	No

▲ Figure 19.31 Design view of the form

When first opened, the form will look like Figure 19.32, with the fields in the lower part of the form hidden.


Create a new order
Create a new order

Order_no (New)

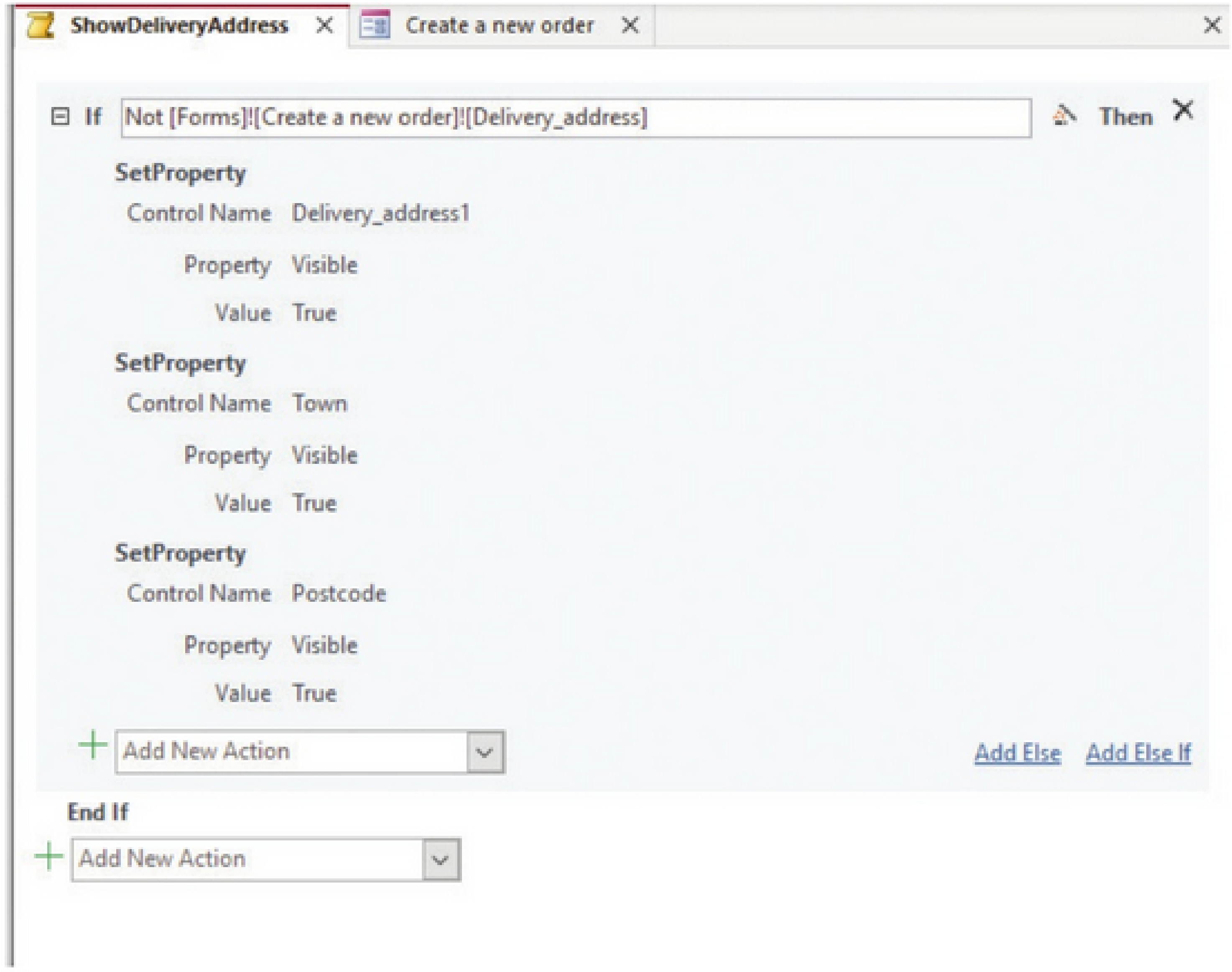
Order_Date 24/01/2024

Customer

Delivery address same as customer address? ☑

▶ Figure 19.32 The form when first opened, the delivery details are not visible

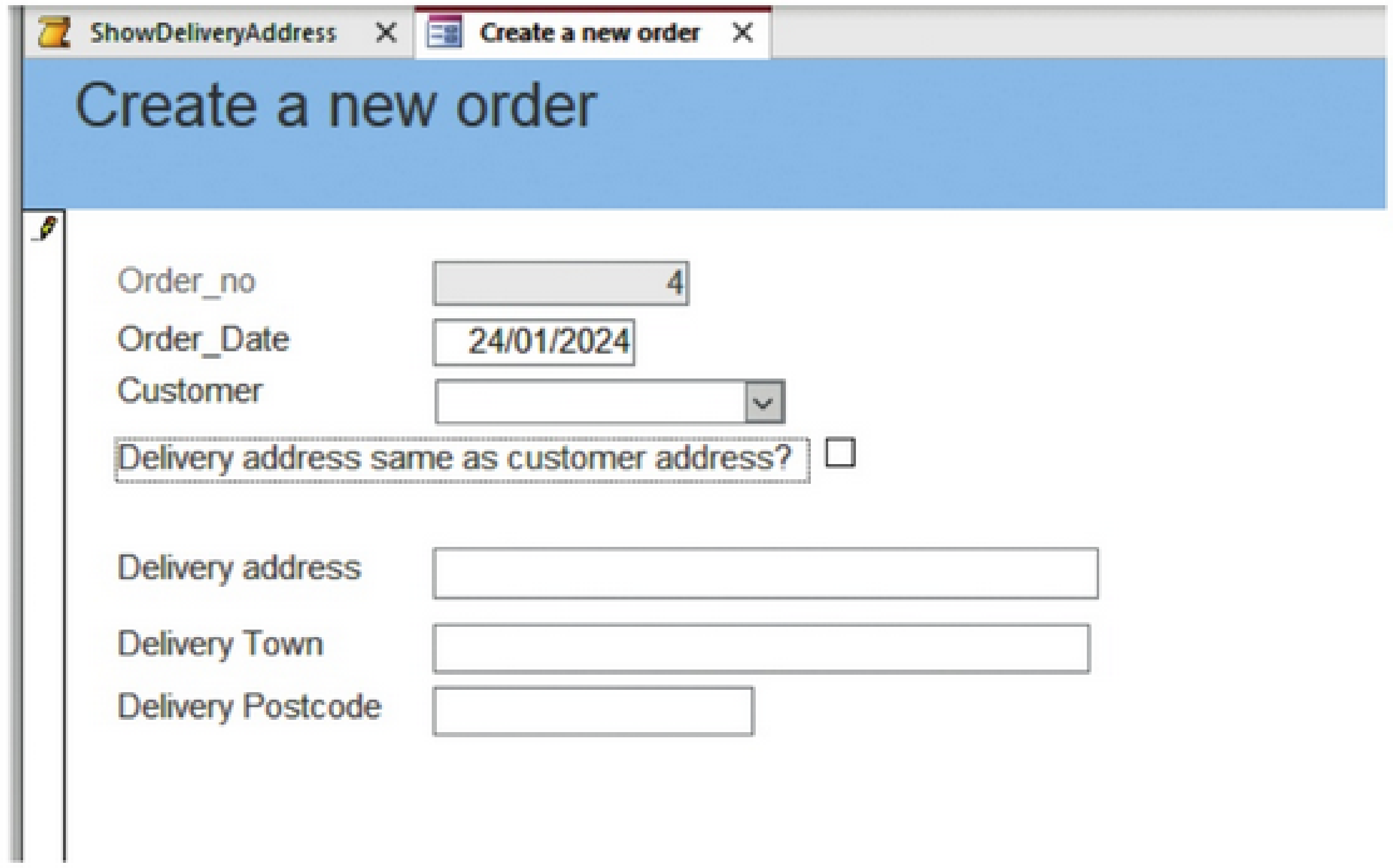
If the delivery address is different then the user needs to uncheck the check box. You will need a macro that is triggered by the On Click event for the check box so that when the box is unchecked the macro will change the visible property for the fields in the lower parts of the form to 'True'. Figure 19.33 shows the macro code to achieve this.


ShowDeliveryAddress
Create a new order
Add Else
Add Else If

```
If Not [Forms]![Create a new order]![Delivery_address]    Then
    SetProperty
        Control Name  Delivery_address1
            Property  Visible
               Value  True
    SetProperty
        Control Name  Town
            Property  Visible
               Value  True
    SetProperty
        Control Name  Postcode
            Property  Visible
               Value  True
  + Add New Action
End If
+ Add New Action
```

▶ Figure 19.33 The macro that makes the delivery address field visible

When the user clicks the checkbox (which by default is checked), this unchecks the box and the macro runs and makes the fields visible. Figure 19.34 shows the form with the fields visible.



Create a new order

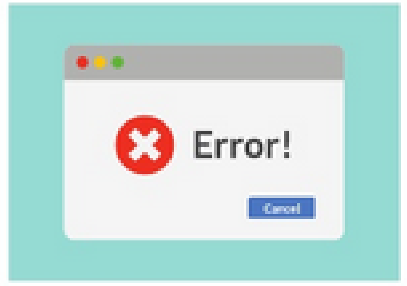
Order_no	4
Order_Date	24/01/2024
Customer	
Delivery address same as customer address?	☐
Delivery address	
Delivery Town	
Delivery Postcode	

▲ Figure 19.34 The form showing the checkbox is not checked so the delivery details are visible

The macro could be further improved by changing the delivery address field's 'visible' property back to false if the checkbox is clicked again, making it checked again.

Appropriate help text. As always, you need to ensure that you use relevant help messages on the form, and you need to clearly identify mandatory and optional fields.

Meaningful error messages. When you add validation to a table or form it is important to complete the 'validation text' property, which defines the text in the error message that pops up should the validation rule be broken. The message should clearly explain to users what input is required to match the validation rule.



▲ This is not a helpful error message

Sensible and meaningful data entry field names. Using the Wizard to create forms and reports makes them easy to create, but the Wizard takes the field name from the table and uses them as labels for the fields on the form. As mentioned in Topic 18, field names used in tables should avoid the use of spaces in them, but this doesn't look good on a form or report. Other abbreviations may also be used in the table fields' names, which need to be expanded when used as a label for a field on a form. Once you have created a form or report using the Wizard you should therefore edit the field names, adding spaces and making sure the labels describe the field in a way that users will be able to understand, avoiding abbreviations and technical terms.

Data entry features. Features such as drop-down (combo) boxes, radio buttons and input masks were covered in Topic 18.

Restricting editing. In some cases you may have fields on a form that you do not want users to be able to change. You have seen this already when you protected the primary key field of a table by setting its Enabled property to 'No'. You also saw how you can change field properties depending on selections that users make, so you could enable a field if, because of another choice a user made, it became relevant for them to be able to edit the field.

Intuitive layout. User interface design is a subject in itself, but there are a few basic things that you can do to help make the interface intuitive.

- Make sure all fields and buttons are clearly labelled.
- Make it clear which data entry fields are mandatory and which are optional. This can be done by placing an asterisk (*) next to the mandatory ones.
- Have a visual hierarchy with the most important things at the top of the form. Use text size to indicate importance, with more important field information in a larger size font.
- Where you have menus that can take users to other forms, always provide a way back to the menu (typically with a back button).
- Provide a way to cancel an operation. Data entry forms, for example, should have a cancel button to allow the user to abort saving the record.

Features of good design. What you should be aiming for here is a neat, professional look to the forms and reports you create. This is also a big topic, but a few basic suggestions are:

- Use a simple colour scheme, with just a few colours – two or maybe three complementary colours that are not too bright.
- As with colours, use a very limited number of fonts, probably the same typeface for all the forms, although you can vary the size.
- Be consistent in your layout when you have several forms. Aim for a 'house style' where the same formatting and layout is used in all the forms in an application.
- Make sure that the layout is neat, with fields aligned to each other and labels aligned with the fields that they belong to.

Always check that the text you have on forms, in the help messages, field names and error messages is free from spelling and grammar errors.

MACROS

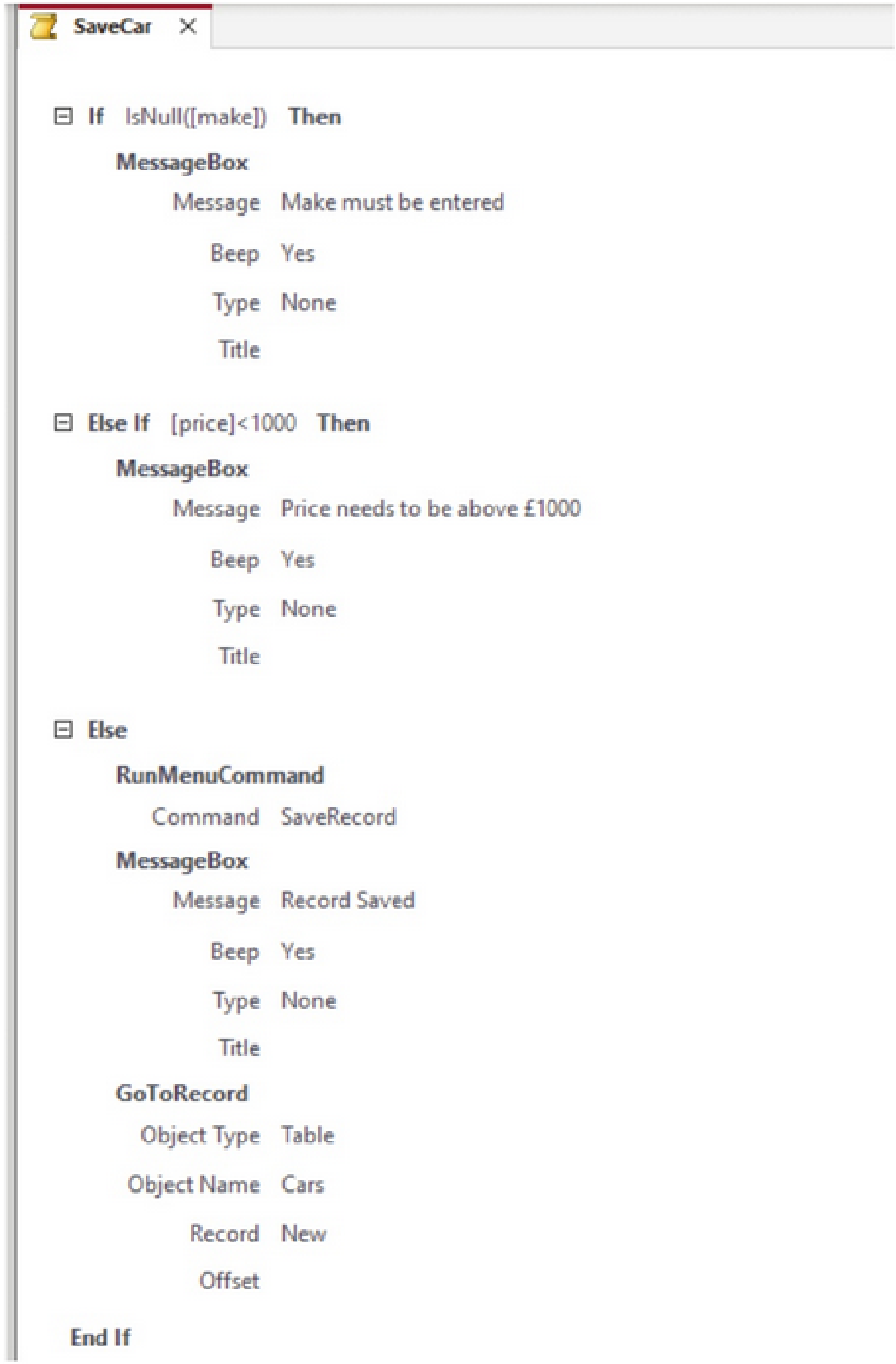
You have already looked at a few macros but there are some features we have yet to cover.

Program flow. You have already used the 'if' instruction but you haven't yet seen how additional conditions can be added. Suppose you wanted to add some field validation to a save button on a form so the save would not work if the data was invalid. You have already created a Save macro for a form to add a car to the Cars for Sale database, which provided a confirmation message that the record was saved. You can now modify this macro to do some validation on the form. For example, you could check that the make of the car has been added by doing a presence check, and that the price is greater than £1000 (because the garage doesn't sell any cars below that price).

The structure for an If…Elseif…Else construct is as follows:

```
If (condition) then
    Instructions to run if condition is true
Elseif (condition) then
    Instructions to run if condition is true
Else
    Instructions to run if neither condition is true
End if
```

The modified macro to carry out the validation before the save is done is shown in Figure 19.35.

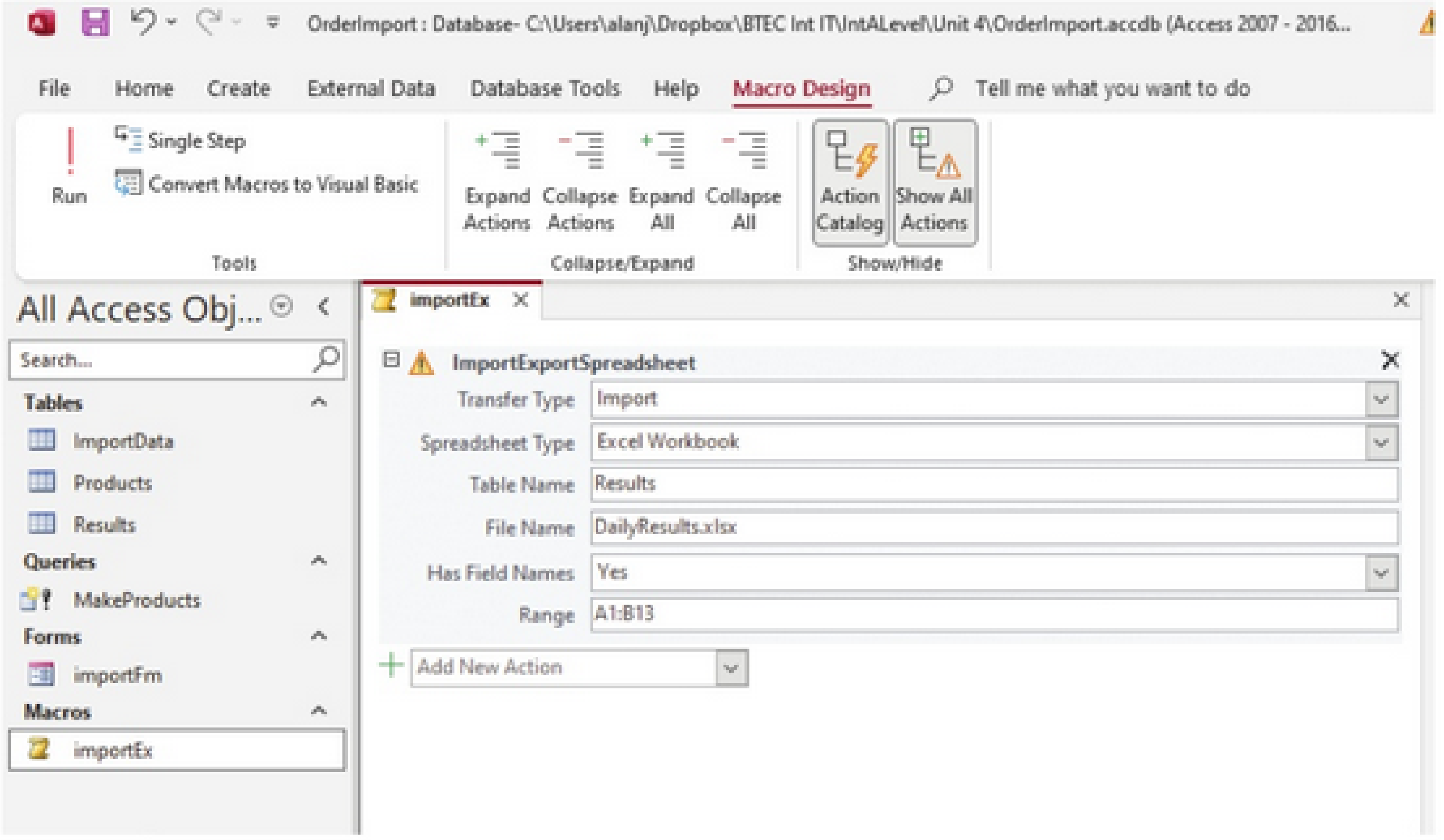


```
If IsNull([make]) Then
    MessageBox
        Message  Make must be entered
        Beep  Yes
        Type  None
        Title
Else If [price]<1000 Then
    MessageBox
        Message  Price needs to be above £1000
        Beep  Yes
        Type  None
        Title
Else
    RunMenuCommand
        Command  SaveRecord
    MessageBox
        Message  Record Saved
        Beep  Yes
        Type  None
        Title
    GoToRecord
        Object Type  Table
        Object Name  Cars
        Record  New
        Offset
End If
```

▶ Figure 19.35 The modified SaveCar macro showing the use of 'if' and 'elseif' instructions

Data import and export. You can use a macro instruction called ImportExportSpreadsheet to import or export an Excel Spreadsheet, although it comes with a warning that it is 'unsafe' as it is possible to hide malicious code in Access and Excel macros. To use this instruction in a macro you need to select the Show all Action icon in the Macro Design toolbar.

Suppose you were developing an application for a scientific database that needed to make regular imports from a computer-controlled machine that provided results in Excel format. To make it easy for the scientists to import the results each day, you need to provide an automated process to do the import. A macro to do this is shown in Figure 19.36.


OrderImport : Database- C:\Users\alanj\Dropbox\BTEC Int IT\IntALevel\Unit 4\OrderImport.accdb (Access 2007 - 2016...

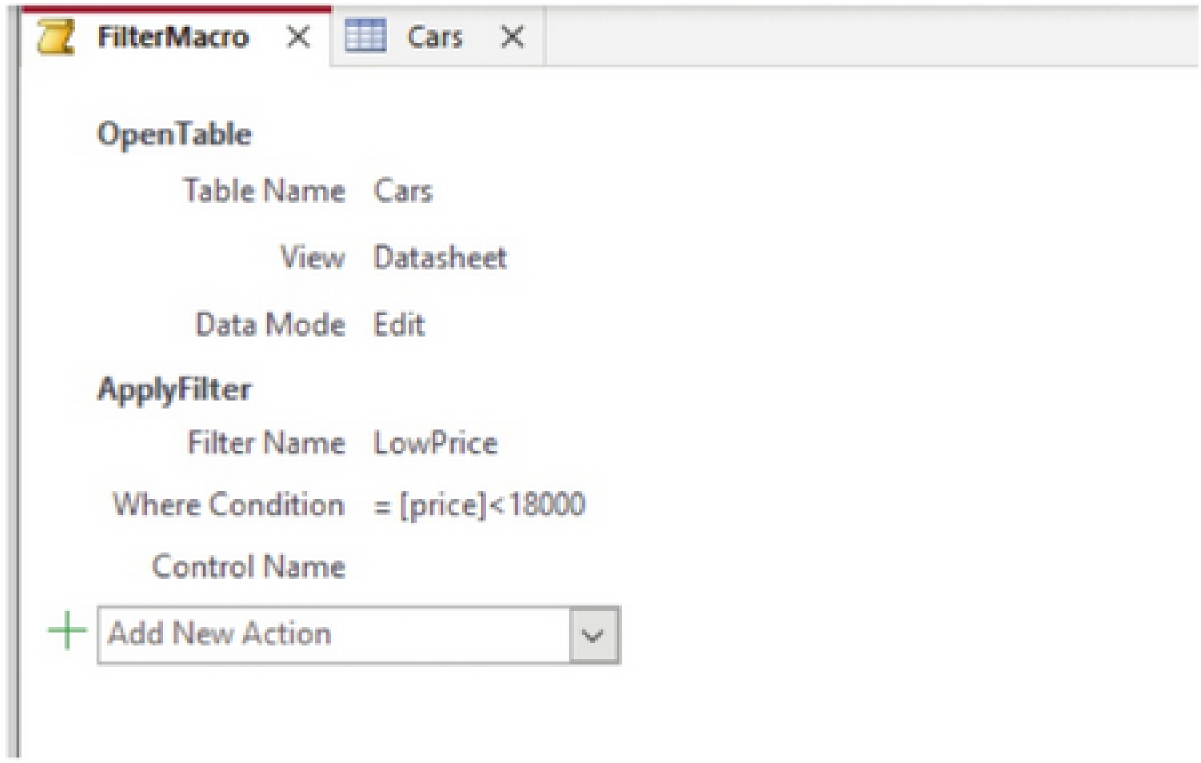
File
Home
Create
External Data
Database Tools
Help
Macro Design
Tell me what you want to do
Run
Single Step
Convert Macros to Visual Basic
Tools
Expand Actions
Collapse Actions
Expand All
Collapse All
Collapse/Expand
Action Catalog
Show All Actions
Show/Hide
All Access Obj...
Search...
Tables
ImportData
Products
Results
Queries
MakeProducts
Forms
importFm
Macros
importEx
importEx
Add New Action

ImportExportSpreadsheet
Transfer Type: Import
Spreadsheet Type: Excel Workbook
Table Name: Results
File Name: DailyResults.xlsx
Has Field Names: Yes
Range: A1:B13

▲ Figure 19.36 A macro to import a spreadsheet file

Performing queries and filtering data. You can include the OpenQuery instruction in a macro to open an existing query. You can also use the ApplyFilter instruction to filter data.

Figure 19.37 shows an example of a macro that applies a filter to a table.


FilterMacro
Cars
Add New Action

```
OpenTable
    Table Name  Cars
    View  Datasheet
    Data Mode  Edit
ApplyFilter
    Filter Name  LowPrice
    Where Condition  = [price]<18000
    Control Name
```

▲ Figure 19.37 A macro that applies a filter to a table

User interface management. You can use macros and attach them to command buttons that carry out various actions on the user interface, such as opening a form (with the OpenForm macro instruction) and closing an object. The CloseWindow instruction can be used to close tables, forms, queries, reports, and so on.

EVALUATING FEATURES OF A USER INTERFACE

To evaluate the features that have been included in a user interface you need to explain:

Why you have included the feature. What was the requirement or problem that made you include the feature? For example, you might have a form where a user needs to select a limited number of options (the requirement) so you have provided a list box that shows the options that the user can select (the feature you have used to meet the requirement).

How the feature benefits the user and the system. In the example above, the use of a list box makes it easy for the user because they don't have to enter the option themselves. If the list of options is quite short, then a list box is a good choice because all the options can be seen and the user just needs to click on the correct one. A combo box would require the user to click the drop down for the box, scroll through the options then select the required one. However, a combo box takes less space on the form so it might be better on a form with a lot of fields. The use of a list box or combo box benefits the system because it prevents invalid entries and makes sure the data saved in the field is consistent (in terms of things like capitalisation and spelling).

SKILLS

ANALYSIS, PROBLEM SOLVING

ACTIVITY 5

1. Create a data entry form for the Book table in the library database. The form should use combo or list boxes to allow the user to select the author and category of the book.
2. Using a macro, add a save button to provide a confirmation message to the user that the book has been saved.
3. Evaluate the data entry forms you have created. Do you think non-IT specialists would find them easy to use? What features could be added to improve their usability?

KEY TERMS

expected result: this is the outcome you expect when testing; if the actual outcome of a test matches the expected outcome then the test is passed

prototype: an early version of a product such as a user interface, which you can show to users to gain feedback on suitability and appropriateness

TESTING THE SOLUTION

When you have completed a database application you need to test it. Testing is important because:

1. You need to check that the application you have created meets the requirements that were defined before the development started. In other words, does it do what it is supposed to do?
2. You need to confirm that the database application you have created works as it should. Does it produce the correct results? This is particularly important where calculations are being done.
3. You need to test that all the features and functions of the application work as they should. For example, do all the buttons produce the **expected result**?

Testing has traditionally been done at the end of the development process, and then the application is presented to the users. Introducing the application to a user at the end of the development process has the risk that the completed application may not meet the needs of the user in the way they had in mind. A much better approach is to show users **prototypes** of the application as it is being developed to gain their feedback. This way, any misunderstanding between the users and developer about what is required can be identified and resolved early in the development process.

CARRYING OUT TESTS

There are several different tests that can be carried out, but for each test you need to create a test plan. The idea behind a test plan is that you:

1. list the test you plan to carry out
2. state what you expect to happen; this way you define what constitutes a successful test
3. carry out the tests you have listed and note the actual results
4. compare the actual results of the test with what you expected to happen.

If the actual results match what you expected to happen then you can say the test has been passed.

If the actual results and the expected results don't match then the test has failed and further investigation is needed.

An example, partly completed, test plan for testing that the menus in an application work correctly might look like Table 19.1.

TABLE 19.1: A MENU TEST PLAN

Test no	Test description	Expected result	Actual result	Comments
1	Main menu: Click the 'Add record' button	The Add record form opens		
2	Main menu: Click the 'Edit record' button	The Edit record form opens		
3	Main menu: Click the 'Order report' button	The Order report opens in print preview mode		
4				

Testing that data validation for input fields works is a little more complex. Suppose you needed to test the validation on a field where the height of adults is recorded. The validation might be that the value must be greater than 1 metre and less than 2.5 metres. Testing should be done using what are called boundary values, so 1 metre is invalid but 1.1 is valid. Tests should also be conducted with no input data and non-numeric data. The test plan should look like Table 19.2.

TABLE 19.2: A VALIDATION TEST PLAN

Test no	Test data	Expected result	Actual result	Comments
1	1	Rejected with an error message displayed		
2	1.1	Accepted		
3	2.4	Accepted		
4	2.5	Rejected with an error message displayed		
5	(null)	Error message 'Height must be entered'		
6	Three	Error message 'Must be a number'		

The actual results are completed when the testing is done and could consist of a written description of what happens or a screenshot showing the result.

Testing output requires that you create a number of 'scenarios' where you create appropriate input data and predict what output that will produce (the expected results), such as what calculated results will be produced. For example, in an order processing system, the test data should consist of a range of orders (with large and small numbers of items ordered). The expected result will include the item totals (item price × quantity), grand total for the order, tax calculations and postage and packing calculations. All of these will need to be manually calculated to work out what the expected values should be.

SKILLS
ANALYSIS

ACTIVITY 6

1. Create a test plan for the data entry form you created for the library database in Activity 5.
2. A database application has a calculation for postage on orders. If the order value is less than £10.00, postage is £3.00. If the value is more than £10.00 but under £25.00 then postage is £5.00. For orders over £25.00, postage is free. Create a test plan to define test data expected results for this calculation.

EVALUATION AND ENHANCEMENT

EVALUATING A DATABASE

The most important question when evaluating any software development project is whether or not it meets the requirements of the users. Every software product will have its strengths and weaknesses, and when evaluating the final product there are a number of things that you need to consider.

Is it fit for purpose? The purpose of the database should have been defined at the outset by those who will use the product. You can consider how well it achieves that purpose. To a certain extent this will depend on how detailed the requirements were in the first place. In the assessment of this unit you will be provided with a scenario that describes the database and user requirements. This scenario will also be extended in some of the questions. To evaluate what you have produced you will need to carefully check if the requirements listed in the scenario have been met by what you have produced.

Is it easy to use? This can be difficult for you, as the creator of the database, to judge. Remember that the users of the database will probably not be IT professionals, so things that are obvious for you may be less so for them. You might find it helpful to show your user interface designs to your peers and, if possible, to others who are not IT experts to get their opinions on how intuitive the interface you are developing is. Of course, you can't do this in the final assessment for this unit, but it might help you get some ideas about what users might find difficult to understand about interfaces.

Are some aspects of the system constrained by the software used? Microsoft Access is designed for use as a desktop database and it lacks features required for multi-user environments and web-based interfaces. With a database, this is particularly limiting as data by its nature is most useful when shared with others.

How easy is it to maintain the database? Would it be easy for someone else to update, correct and add to your database? The ease of maintenance of an Access database will depend upon how detailed and accurate your design documentation is. This documentation should allow someone else to understand how your database is designed and created.

Enhancing a database. After you have evaluated a database it is likely that you will come up with some areas that could be improved or developed. Ideally you will be able to provide realistic suggestions as to how these improvements could be implemented. These might, for example, simply be suggestions for how the user interface could be improved, perhaps by adding clearer instructions, rearranging fields on a form into a more logical order or rewording error messages.

SKILLS
CRITICAL THINKING

ACTIVITY 7

Take a look at one of the databases you have created in the activities, for example, the library database. Write a report on how well you think it would meet the needs of a real library, like the one in your college or school. What other features do you think would be needed? Consider how easy to use the forms and menus you created are. What features could you add to improve their usability?

THINKING LIKE AN IT SPECIALIST

Computers, the internet, modern life: they all depend on data. Because of this, database technology is of central importance to information technology. Relational databases are structured using tables and relationships, but this rigid structure does not suit all types of data. Also, some applications (such as social media and big data) need to store vast amounts of data and be able to process it quickly. This has given rise to a different type of database known as a noSQL database. A noSQL database doesn't use tables and is suited to data that is not so clearly structured. Doing some research into noSQL databases will help you understand them better: the different types of noSQL methods that are used in databases, their advantages and disadvantages when compared with traditional relational databases and the applications that use noSQL databases.

EXAM TIP

The exam for this unit is a practical task. You will be given an existing database and asked to add new features to it, such as creating forms, queries and reports. You may be asked to import some data into the database from a text or spreadsheet file. Therefore, it is important that you practise and develop your skills in using Access to complete these tasks.

CHECKPOINT

1. What issues need to be considered before deleting a record on a related table?
2. What is the difference between a filter and a query?
3. How would you use a query to find all the records that have a value in a field beginning with the letter 'A'?
4. Suppose you had two fields named 'Price' and 'Postage', how would you add them together in a query and display them in a query column called 'Total'?
5. What is a CSV file?
6. In a form list box what is the property that tells you which item in the list the user has selected?
7. What macro action can be used to display a confirmation message to the user?
8. Which property can be used to hide a form object such as a field from the user?
9. Which property can be used to prevent a user for entering any data in a form field?
10. In the 'If' macro action, what are the 'Elseif' and 'Else' actions used for?

EXAM PRACTICE

Erik runs a cycling club. He wants to use a database to store details of riders' times on different routes. The club has members and riders and they ride on different routes. Erik has been using a spreadsheet to store the data for the club. An extract from the spreadsheet is shown here:

Route ID	Route name	Route distance (km)	Ride ID	Date	Time (mins)	Member ID	Member name	Member age	Member gender
1	Hilly loop	46	1	20/1/24	102	2	Patel	26	M
1	Hilly loop	46	2	20/1/24	98	3	Babatude	38	M
1	Hilly loop	46	3	20/1/24	112	4	Shah	22	F
2	Flat circuit	58	4	27/1/24	216	1	Roberts	19	F
2	Flat circuit	58	5	27/1/24	225	2	Patel	26	M
2	Flat circuit	58	6	27/1/24	230	3	Babatude	38	M
2	Flat circuit	58	7	27/1/24	218	4	Shah	22	F
2	Flat circuit	58	8	27/1/24	221	5	Begum	24	M
3	To the lake	39	9	3/2/24	80	3	Babatude	38	M
3	To the lake	39	10	2/2/24	92	5	Begum	24	M

1. Create a related set of tables to store this data in an Access database. Remember:
   - one member can go on many different rides
   - each ride is on a single route
   - each route could have many rides on it.

   Provide screen shots of each table in Design view and the relationships ERD. [5 marks]
2. Erik wants a data entry form to allow new members to be added. The form should:
   - open at a blank record, ready for data entry
   - use a select (radio) button to allow the member's gender to be recorded
   - validate that the member has an entry in the name field and their age is from 18 to 100
   - have a save button that saves a record only if it has been successfully validated, displays a confirmation message and clears the form for a new record.

   Provide screen prints of your form in Design view and Form view. [8 marks]
3. A data entry form is also needed to allow a new ride to be added. The form should:
   - show today's date as a default in the date field
   - provide a combo or list box to look up the member who did the ride
   - provide a combo or list box to look up the route of the ride
   - have a save button that saves a record, displays a confirmation message and clears the form for a new record
   - have a cancel button to abort the save of the record.

   Provide screen prints of your form in Design view and Form view. [8 marks]
4. Create a query that will allow Erik to enter a route ID and see a list of members and their times on that route.
   - the query should display the route name, the member name and the time taken
   - the times should be sorted with the lowest time first.

   Provide screen prints of your form in Design view and Form view. [3 marks]
5. Create a query that lists each route, a count of the number of times the route has been ridden and the average time across all the members to complete the route. The query should display the route name, route distance as well as the number of times it has been ridden and the average speed.

   Provide screen prints of your query in Design view and Datasheet view. [3 marks]
6. Create a report that lists each member and the rides they have done. Add a column or bar chart to the report that shows, for each member, the times they achieved on each ride. Make sure all the field labels and contents can be seen. The report should have suitable labels and titles and should fit on one page.

   Provide screen prints of your report in Design view and Print Preview. [4 marks]

GLOSSARY

4G/5G fourth and fifth generation technology standard for cellular networks

accessibility making an application accessible to as many people as possible, including those with disabilities

algorithm a finite set of predefined steps to solve a problem

application programming interface (API) a set of functions that enable two computer systems or programs to communicate and share data

archive transfer of less frequently used data to a backing store

artificial intelligence (AI) a computer science/IT discipline that aims to build computer systems that can replicate human intelligence

atomic data data that represents a single attribute and cannot be broken down further

back-office processes administrative and logistical tasks that support a business but do not interact directly with the customer/client

bandwidth the amount of data that can be transmitted over a connection in a given amount of time

brute-force attack persistent and repeated attempts to gain access to a system by guessing a password

business-critical data, information or tasks/operations that are essential for an organisation to continue to operate

ciphertext the encrypted version of a message/data

comma separated value (CSV) a text data file where the data values are separated by commas

composite key a key that is made up of two or more fields

criteria a definition of how a query should select data to be displayed

dashboard an interface that allows a user to choose various ways of presenting relevant information

data governance systems and procedures for ensuring data is reliable and secure

data migration extracting data from an existing dataset and transferring it to another

data privacy the protection of data from unauthorised access. Ensures that only those who have permission and a valid need to see the data are able to access it

data redundancy the unnecessary replication of data in a database

datatype a field property that sets the sort of data that can be stored in the field

database management system (DBMS) a type of software application that provides tools for creation and manipulation of databases and the data that they hold

e-commerce the buying and selling of goods and services using computer systems, such as over the internet

ecosystem a set of tools, libraries and applications designed for a particular purpose or platform e.g. Apple, Android

encapsulate to group or bundle together related data and/or functions

entity relationship diagram (ERD) a diagram used by database designers to show the tables in their database and the links between them

e-waste discarded electrical equipment such as computers, smartphones and televisions

expected result this is the outcome you expect when testing; if the actual outcome of a test matches the expected outcome then the test is passed

feature creep continual addition of new features to a product in development that affects the team's ability to complete the project effectively and negatively affects deadlines and budgets

foreign key a column or columns in a table that links to unique data in another table. This is often a primary key in the other table

graphical user interface (GUI) an interface that uses icons, buttons and menus with a mouse or touch-based pointing method

haptic a form of computer interaction based on touch

HTTPS protocol the secure version of the protocol used to transfer and render webpages

hypervisor a program that runs and manages virtual machines

incremental small but measurable developments and changes that move a project closer to its completion

inference engine the decision-making part of an expert system

infrastructure hardware and software resources that support data and processes

intuitive an interface that can be navigated by instinct and does not require instruction

IoT (internet of things) interconnection of everyday objects through the internet to allow the sending and receiving of data

iterative an approach to project/system development where outcomes are continually reviewed and improved

knowledge base a large repository of information related to a specific area of expertise

latency the delay between requesting and receiving data

lookup a method of providing input data by either looking it up from another table or by providing a list

machine learning a subset of artificial intelligence. The use of algorithms whose improves as they are exposed to more data

macro a series of actions that allow you to automate database procedures such as saving a record

metadata details or information that describe other data to provide deeper meaning or understanding

mitigation an action or strategy intended to reduce the seriousness or harm caused by a potential risk

network diagram a diagram that provides visual representation of project tasks, dependencies and prerequisites (e.g. arrow diagram method, Gantt chart)

node a key point in a network relating to a specific connection or device

one-to-many relationship where one record in a table is related to many records in another table

plain text the original and/or decrypted version of a message/data

portability the ease with which computing services can be transferred between systems and providers

primary key a code, number or name that uniquely identifies a single record in a database

process automation the use of technology to automate complex, repetitive or mundane tasks

project lifecycle the main phases of a project (e.g. initiation, planning, execution, monitoring, closure)

project manager the individual responsible for overseeing and delivering a project by managing human and physical resources

prototype an early version of a product such as a user interface, which you can show to users to gain feedback on suitability and appropriateness

quality assurance (QA) the process of ensuring that the desired level of performance, design and reliability is maintained

query a database tool that filters data to match given criteria

record an individual row in a data table that contains data about a single person, item or object

referential integrity a setting in Microsoft Access that prevents you creating a record in a linked table at the 'many' end using a foreign key that does not exist in the table at the 'one' end

relational database a database made up of multiple related tables

secure socket layer (SSL) a networking protocol that provides secure connections between web clients and web servers

silo a collection of data used and controlled by one group, department or business that is not accessible by the rest of the organisation

snapshot a saved copy of a system, including all data and settings at a given point in time

stakeholder an individual or group with a particular interest in or link to a project

structured query language (SQL) a widely used programming language used for manipulating and maintaining data

stereoscopic combining two images (still or moving) taken at slightly different angles to give the illusion of depth

supervised learning the use of labelled data to train a machine learning model

target audience the typical users (audience) that an application is aimed at

telematics an in-vehicle system that uses GPS and vehicle diagnostics to provide detailed, real-time information

thin client a computer that runs resources stored on a remote server rather than locally installed resources

threat actors also known as malicious actors or cyberthreat actors. Individuals or groups that intentionally target digital systems to cause damage, disruption or for financial gain

transaction a sale, where a customer exchanges money for goods or services

unsupervised learning the use of unlabelled data to train a machine learning model

user experience (UX) how a user experiences their interaction with a system

validation techniques and processes designed to ensure data is reasonable and follows specific rules

version control tracking and managing changes in documents or software

visual hierarchy the order in which elements are noticed and their perceived importance

workflow a sequence of tasks or processes carried out by individuals, groups or organisations to complete an objective

ACKNOWLEDGEMENTS

All Credits

The author and publisher would like to thank the following individuals and organizations for permission to reproduce photographs.

(key: b-bottom, c-centre, l-left, r-right, t-top)

Prominent Image Credit(s):

Front Cover: SEAN GLADWELL/Moment/Getty images; **Back Cover:** MirageC/MomentGetty images

Non-Prominent Image Credit(s):

123RF GB LIMITED: adam121 118, rawpixel 129, 221, Marina Basalaeva 229; **Alamy images:** Cavan Images 89; **Shutterstock:** jamesteohart 1, Gorodenkoff v, 2, SvedOliver 6, leo_photo 7, Summit Art Creations 8, vectortatu 9, Fractal Pictures 11, Rawpixel.com 15, Maksim Shmeljov 16, East pop 24, metamorwork 31, 59, PromKaz 33t, junpiiiiiiiiiiii 33b, David Jancik 34t, dencg 34c, Sunshine Seeds 34b, BritCats Studio 35t, Ninetechno 35b, New Africa 36t, Andrey_Popov 36c, 63, 79, 94, Magnetic Mcc 36b, Ohmega1982 37, Jelena Stanojkovic 46, Pixelbliss 47, Joanna W Photo 49, ymgerman 50, Antonio Guillem 51t, Ground Picture 51b, Tatomirov 55, Andrea Danti 60, ivector 64, Marta Sher 65, Halfpoint 68, elenabsl 69, 126, Jiw Ingka 70, HL12 72, vectorstudi 75, voronaman 80, Yurich 92, Zapp2Photo 95, Tee11 108, EPStudio20 114, chayanuphol 115, Morten B 117, Song_about_summer 120, EgudinKa 128, Nebojsa Panchenko Vladimir 134, Luca Lorenzelli 144, Leonardo da 154, Kateryna Reka 157, jamie cross 158, stickasa 177, Argus 194, ShlyahovaYulia 195, McLittle Stock 222, stefanphotozemun 223, alexdndz 225, U-STUDIOGRAPHY DD59 226, Pong Ch 233, BestForBest 238, Primakov 239, Zita 242, maicasaa 243, Aleksandr Finch 246, stockcreations 250,